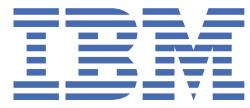


SPSS Modeler



Tables of Contents

IBM SPSS Modeler V18.4.0 documentation	1
Installation	
Client Installation (Concurrent User License)	
Installation instructions	1
System requirements	1
Installing	1
Installing from a downloaded file	2
Silent installation	2
Notes for installation	3
Licensing your product	3
Using the license authorization wizard	3
Troubleshooting an installation	
Invalid digital signature on installation	3
Configuring IBM SPSS Modeler to work with IBM SPSS Statistics	4
Database Access	4
Checking out/in a commuter license	5
Applying fix packs	5
Uninstalling	5
Client Installation (Authorized User License)	
Installation instructions	5
System requirements	6
Authorization code	6
Installing	6
Installing from a downloaded file	6
Silent installation	6
Notes for installation	7
Licensing your product	7
Using the license authorization wizard	8
Viewing your license	8
Troubleshooting an installation	
Invalid digital signature on installation	8
Configuring IBM SPSS Modeler to work with IBM SPSS Statistics	8
Database Access	9
Applying fix packs	9
Uninstalling	9
Updating, modifying, and renewing IBM SPSS Modeler	10
Client Installation (Mac)	
Installation overview	10
System requirements	10
License types	10
Installing	10
Installing from a downloaded file	10
Notes for installation	10
Authorized user license installation	11
Authorization code	11
Licensing your product	11
Using the license authorization wizard	11
Viewing your license	11
Updating, modifying, and renewing SPSS Modeler	11
Concurrent user license installation	
Using the license authorization wizard	12
Checking out/in a commuter license	12
After installation	12
Configuring IBM SPSS Modeler to work with IBM SPSS Statistics	13
Database Access	13
Applying fix packs	14
Uninstalling	14
Concurrent User License Administrator's Guide	
Administrator's guide	14

Before you start	14
Citrix and Terminal Services	15
Mixed licensing	15
Installing the concurrent license manager	15
Upgrading the license manager	16
Installing the license manager on Windows	16
Installing the license manager on Mac OS	16
Installing the license manager on non-Windows systems	16
Installing the license manager administrator	16
Licensing your product	17
Installing a license in a virtual environment	17
Installing a license from the command prompt	18
Using licenseactivator to install a license automatically	18
Installing a license manually	19
Adding a license	19
Viewing your license	19
Testing the license manager	20
Installing the product on the local desktop computers	20
Pushing an installation to Windows computers	20
Uninstalling a Previous Version	20
Properties for push installations	21
MSI files	21
Command line example	21
Using SMS to push the installation	21
Using Group Policy or related technology to push the installation	21
Pushing an uninstallation	21
Administering the concurrent license	22
Starting the WlmAdmin Application	22
Adding a server	22
Obtaining log information	23
Viewing details about a license	23
Setting up redundant license servers	23
Configuring commuter licenses	24
Configuring license reservations	25
Starting and stopping the license manager	26
Configuring the license manager to start automatically	26
Uninstalling the license manager	27
Uninstalling the license manager administrator	27
Troubleshooting Desktop Computers	27
Running lswhere	27
Authorized User License Administrator's Guide	
Administrator's guide	27
Before you start	28
Citrix and Terminal Services	28
Installing the product on the local desktop computers	28
Pushing an installation to Windows computers	28
Uninstalling a Previous Version	28
Properties for push installations	29
MSI files	29
Command line example	29
Using SMS to push the installation	29
Using Group Policy or related technology to push the installation	29
Pushing an uninstallation	30
Using licenseactivator	30
License File	31
Modeler Scoring Adapter Installation	
IBM SPSS Modeler Scoring Adapter Installation	31
IBM SPSS Modeler scoring adapter installation	31
About scoring	31
Migrating to a new version	32
Installing IBM SPSS Modeler Server Scoring Adapter for Netezza	32
Installing IBM SPSS Modeler Server Scoring Adapter for Teradata	33

Installing IBM SPSS Modeler Server scoring adapter for Db2 LUW	34
Modeler Server for UNIX Installation Instructions	
Installation instructions	35
System requirements	35
Additional requirements	35
Installing	36
Graphical installation wizard	36
Command line installation	37
Silent installation	37
After You Install IBM SPSS Modeler Server	37
Installing IBM SPSS Modeler Batch	38
Configuring IBM SPSS Modeler to Work with IBM SPSS Statistics	38
Enabling IBM SPSS Statistics Programmability	39
Starting the Process	39
Checking the Server Status	39
Connecting End Users	39
IBM SPSS Data Access Pack Technology	39
Configuring IBM SPSS Modeler Server for Data Access	40
Uninstalling	40
Modeler Server for Windows Installation Instructions	
Installation instructions	41
System requirements	41
Installing	41
Destination	41
Silent installation	42
IP Address and Port Number	42
Troubleshooting an installation	
Invalid digital signature on installation	42
After You Install IBM SPSS Modeler Server	43
Installing IBM SPSS Modeler Batch	43
Configuring IBM SPSS Modeler to Work with IBM SPSS Statistics	43
Checking the Server Status	44
Connecting End Users	44
IBM SPSS Data Access Pack Technology	44
Uninstalling	44
Modeler Premium Installation	
IBM SPSS Modeler Premium component overview	45
Installing IBM SPSS Modeler Premium Client	45
System requirements	45
Installing	45
Installing from a downloaded file	46
Installing from a network location	46
Silent installation	46
After you install SPSS Modeler Premium	47
Removing IBM SPSS Modeler Premium	47
Installing IBM SPSS Modeler Premium Server	47
System requirements	48
Installing	45
Installing on Windows systems	48
Installing on UNIX systems	48
Silent installation	48
Removing IBM SPSS Modeler Premium Server	50
Removing from Windows systems	50
Removing from UNIX systems	50
Essentials for R	
IBM SPSS Modeler - Essentials for R: Installation Instructions	50
Overview	50
Install the IBM SPSS Modeler application	51
Download and install R	51
Download and install IBM SPSS Modeler - Essentials for R	53
Install IBM SPSS Modeler - Essentials for R for Windows	53
Install IBM SPSS Modeler - Essentials for R for UNIX	54
IBM SPSS Modeler - Essentials for R for Mac	54

Silent installation	54
Running Extension nodes in IBM SPSS Modeler Solution Publisher and IBM SPSS Collaboration and Deployment Services	55
Repairing an installation	56
Uninstalling IBM SPSS Modeler - Essentials for R components	56
Windows	56
UNIX	56
Deployment Adapter Installation - Installation Manager	
Installing IBM SPSS Modeler Server Adapter	56
About IBM SPSS Modeler Server Adapter installation	56
System requirements	57
Installation files	57
Getting started with Installation Manager	57
Repository preferences	58
Setting repository preferences in wizard mode	58
Setting repository preferences in console mode	59
Passport Advantage preferences	59
Setting Passport Advantage preferences in wizard mode	59
Setting Passport Advantage preferences in console mode	60
Installing IBM SPSS Modeler Adapter	60
Installing in wizard mode	61
Installing in console mode	62
Installing silently by using a response file	63
Configuring the adapter for IBM SPSS Collaboration and Deployment Services Web services on Linux	64
Configuring the adapter for SPSS Statistics	64
Troubleshooting	64
Uninstalling IBM SPSS Modeler Server Adapter	64
Uninstalling by using wizard mode	65
Uninstalling by using console mode	65

IBM SPSS Modeler Help

User's Guide	
About IBM SPSS Modeler	66
IBM SPSS Modeler Products	66
IBM SPSS Modeler	66
IBM SPSS Modeler Server	66
IBM SPSS Modeler Administration Console	67
IBM SPSS Modeler Batch	67
IBM SPSS Modeler Solution Publisher	67
IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services	67
IBM SPSS Modeler Editions	67
Documentation	68
SPSS Modeler Professional Documentation	68
SPSS Modeler Premium Documentation	68
Application examples	69
Demos Folder	69
License tracking	69
What's new in version 18.4.0?	69
Lists and geospatial data	70
Product overview	70
Getting started	70
Data Mining and Modeling	71
About Predictive Analytics	71
Modeling Techniques	71
How Do I Know Which Technique To Use?	72
How IBM SPSS Modeler Can Help You	72
Where You Can Use IBM SPSS Modeler	72
How IBM SPSS Modeler works	73
How to Use IBM SPSS Modeler	73
Starting IBM SPSS Modeler	73
Launching from the Command Line	74
Connecting to IBM SPSS Modeler Server	74
Adding and Editing the IBM SPSS Modeler Server Connection	75
Searching for Servers in IBM SPSS Collaboration and Deployment Services	75

Connecting to Analytic Server	76
Changing the temp directory	76
Starting Multiple IBM SPSS Modeler Sessions	77
IBM SPSS Modeler Interface at a Glance	77
IBM SPSS Modeler Stream Canvas	78
Nodes palette	78
IBM SPSS Modeler Managers	78
IBM SPSS Modeler Projects	79
IBM SPSS Modeler Toolbar	79
Customizing the Toolbar	80
Customizing the IBM SPSS Modeler window	80
Changing the icon size for a stream	80
Using the Mouse in IBM SPSS Modeler	81
Using shortcut keys	81
Printing	82
Automating IBM SPSS Modeler	82
Understanding data mining	83
Types of models	83
Data Mining Examples	86
Building streams	86
Stream-building overview	86
Building data streams	86
Working with nodes	87
Adding Nodes to a Stream	87
Connecting Nodes in a Stream	87
Bypassing Nodes in a Stream	88
Disabling Nodes in a Stream	89
Adding Nodes in Existing Connections	89
Deleting Connections between Nodes	90
Setting options for nodes	90
Caching options for nodes	90
Previewing data in nodes	91
Locking nodes	92
Working with streams	92
Setting options for streams	92
Setting general options for streams	93
Setting date and time options for streams	94
Setting number format options for streams	95
Setting optimization options for streams	95
Setting SQL logging and record status options for streams	96
Setting layout options for streams	96
Analytic Server stream properties	97
Setting geospatial options for streams	97
Selecting geospatial coordinate systems	98
Viewing stream operation messages	99
Viewing node execution times	99
Setting Stream and Session Parameters	99
Specifying Runtime Prompts for Parameter Values	100
Specifying Value Constraints for a Parameter Type	101
Stream deployment options	102
Looping Execution for Streams	102
Viewing Global Values for Streams	102
Searching for Nodes in a Stream	102
Renaming streams	103
Stream descriptions	103
Previewing stream descriptions	104
Exporting Stream Descriptions	104
Running streams	104
Working with Models	105
Adding Comments and Annotations to Nodes and Streams	105
Comments	106
Operations Involving Comments	106

Listing Stream Comments	108
Converting Annotations to Comments	109
Annotations	109
Saving data streams	110
Saving States	110
Saving Nodes	110
Saving multiple stream objects	110
Saving Output	111
Encrypting and Decrypting Information	111
Loading files	111
Mapping Data Streams	112
Mapping Data to a Template	112
Mapping between Streams	113
Specifying Essential Fields	113
Examining Mapped Fields	113
Tips and Shortcuts	114
Building charts	114
Chart catalog	115
Layout and terms	115
Building a chart from the chart type gallery	116
Chart types	116
3D charts	116
Bar charts	117
Box plots	119
Bubble charts	120
Candlestick charts	120
Circle packing charts	121
Custom charts	122
Dendrogram charts	122
Dual Y-axes charts	123
Error bar charts	124
Evaluation charts	125
Heat map charts	126
Histogram charts	127
Line charts	128
Map charts	129
Map service options	130
Math curve charts	131
Multi-chart charts	132
Multiple series charts	133
Parallel charts	134
Pareto charts	134
Pie charts	135
Population pyramid charts	136
Q-Q plots	136
Radar charts	138
Relationship charts	138
Scatter plots and dot plots	139
Scatter matrix charts	140
Series array charts	141
Sunburst charts	142
t-SNE charts	143
Time plots	143
Theme River charts	144
Tree charts	145
Treemap charts	146
Word cloud charts	147
Global visualization preferences	147
Dashboard	148
Working with output	149
Viewer	149
Showing and hiding results	149

To hide tables and charts	150
To hide procedure results	150
Moving, deleting, and copying output	150
Moving output in the Viewer	150
Deleting output in the Viewer	151
Changing initial alignment	151
Changing alignment of output items	151
Viewer outline	151
Collapsing and expanding the outline view	152
Changing the outline level	152
To change the size of outline items	152
To change the font in the outline	152
Editing and adding items to the Viewer	152
Adding a title or text	153
To add a text file	153
Pasting objects into the Viewer	153
Finding and replacing information in the Viewer	153
Copying output into other applications	154
Interactive output	155
Export output	155
HTML options	156
Web report options	157
Word options	157
Excel options	158
PowerPoint options	158
PDF options	159
Text options	160
Images only options	160
Graphics format options	161
JPEG chart export options	161
BMP chart export options	161
PNG chart export options	161
EMF and TIFF chart export options	161
EPS chart export options	162
Viewer printing	162
To print output and charts	163
Print Preview	163
Page Attributes: Headers and Footers	163
To insert page headers and footers	164
Page Attributes: Options	164
To change printed chart size, page numbering, and space between printed Items	164
Saving output	164
Saving a Viewer document	165
Pivot tables	165
Pivot tables	166
Manipulating a pivot table	166
Activating a pivot table	166
Pivoting a table	166
Changing display order of elements within a dimension	167
Moving rows and columns within a dimension element	167
Transposing rows and columns	167
Grouping rows or columns	167
Ungrouping rows or columns	168
Rotating row or column labels	168
Sorting rows	168
Inserting rows and columns	168
Controlling display of variable and value labels	169
Changing the output language	169
Navigating large tables	169
Undoing changes	170
Working with layers	170
Creating and displaying layers	170

Go to layer category	170
Showing and hiding items	170
Hiding rows and columns in a table	171
Showing hidden rows and columns in a table	171
Hiding and showing dimension labels	171
Hiding and showing table titles	171
TableLooks	172
To apply a TableLook	172
To edit or create a TableLook	172
Table properties	172
To change pivot table properties	173
Table properties: general	173
Set rows to display	173
Table properties: notes	174
Table properties: cell formats	174
Table properties: borders	175
Table properties: printing	175
Cell properties	175
Cell properties: Font and background	176
Cell properties: Format value	176
Cell properties: Alignment and margins	176
Footnotes and captions	176
Adding footnotes and captions	177
To hide or show a caption	177
To hide or show a footnote in a table	177
Footnote marker	178
Renumbering footnotes	178
Editing footnotes in legacy tables	179
Footnote font and color settings	179
Data cell widths	179
Changing column width	180
Displaying hidden borders in a pivot table	180
Selecting rows, columns, and cells in a pivot table	180
Printing pivot tables	181
Controlling table breaks for wide and long tables	181
Creating a chart from a pivot table	182
Legacy tables	182
Options	182
General options	183
Setting general options	183
Viewer options	183
Setting Viewer options	184
Controlling the initial display state for new output	184
Changing the initial alignment of new results	184
Pivot table options	184
Setting the TableLook for new pivot tables	185
Controlling column width for new pivot tables	185
Creating a custom default TableLook	185
Output options	186
Setting output options	186
Handling missing values	186
Overview of Missing Values	186
Handling Missing Values	187
Handling Records with Missing Values	187
Handling Fields with Missing Values	187
Handling Records with System Missing Values	188
Imputing or Filling Missing Values	190
CLEM Functions for Missing Values	190
Building CLEM expressions	191
About CLEM	191
CLEM Examples	191
Values and Data Types	192

Expressions and Conditions	193
Stream, Session, and SuperNode Parameters	194
Working with Strings	194
Handling Blanks and Missing Values	195
Working with Numbers	195
Working with Times and Dates	196
Summarizing Multiple Fields	196
Working with Multiple-Response Data	197
The Expression Builder	198
Accessing the Expression Builder	198
Creating Expressions	198
Selecting functions	199
Database functions	199
Selecting fields, parameters, and global variables	201
Viewing or selecting values	201
Checking CLEM expressions	202
Find and Replace	202
CLEM language reference	203
CLEM Reference Overview	203
CLEM Datatypes	204
Integers	204
Reals	205
Characters	205
Strings	205
Lists	206
Fields	206
Dates	206
Time	207
CLEM Operators	207
Functions reference	208
Conventions in Function Descriptions	209
Information Functions	210
Conversion Functions	210
Comparison Functions	211
Logical Functions	212
Numeric Functions	213
Trigonometric Functions	214
Probability Functions	214
Spatial functions	214
Bitwise Integer Operations	215
Random Functions	216
String Functions	216
SoundEx Functions	219
Date and Time Functions	219
Converting Date and Time Values	221
Sequence functions	222
Global Functions	225
Functions Handling Blanks and Null Values	225
Special Fields	226
Using IBM SPSS Modeler with a repository	226
About the IBM SPSS Collaboration and Deployment Services Repository	226
Storing and deploying repository objects	227
Connecting to the Repository	228
Entering Credentials for the Repository	228
Browse for repository credentials	229
Browsing the Repository Contents	229
Storing Objects in the Repository	229
Setting Object Properties	230
Choosing the Location for Storing Objects	230
Adding Information About Stored Objects	230
Assigning Topics to a Stored Object	231
Setting Security Options for Stored Objects	231

Adding a User to the Permissions List	231
Modifying Access Rights for an Object	232
Storing Streams	232
Storing Projects	232
Storing Nodes	232
Storing Output Objects	233
Storing Models and Model Palettes	233
Retrieving Objects from the Repository	233
Choosing an Object to Retrieve	234
Selecting an Object Version	234
Searching for objects in the repository	234
Modifying Repository Objects	235
Creating, Renaming, and Deleting Folders	236
Locking and Unlocking Repository Objects	236
Deleting Repository Objects	236
Managing Properties of Repository Objects	237
Viewing Folder Properties	237
Viewing and Editing Object Properties	237
Managing Object Version Labels	238
Deploying streams	239
Stream Deployment dialog	239
Stream Deployment Options	240
Scoring and modeling parameters	241
The Scoring Branch	241
Identifying the Scoring Branch for Deployment	241
Model Refresh	242
How the Refresh Model is Selected	242
Checking a scoring branch for errors	243
Saving streams to IBM Cloud Pak for Data	243
Exporting to external applications	244
About Exporting to External Applications	244
Opening a Stream in IBM SPSS Modeler Advantage	244
Projects and reports	245
Introduction to Projects	245
CRISP-DM View	245
Setting the Default Project Phase	246
Classes View	246
Building a Project	246
Creating a New Project	247
Adding to a Project	247
Transferring Projects to the IBM SPSS Collaboration and Deployment Services Repository	248
Setting Project Properties	248
Annotating a Project	248
Folder Properties and Annotations	249
Object Properties	249
Closing a Project	249
Generating a Report	250
Saving and Exporting Generated Reports	251
Customizing IBM SPSS Modeler	251
Customizing IBM SPSS Modeler options	251
Setting IBM SPSS Modeler options	252
System Options	252
Managing Memory	252
Setting Default Directories	253
Setting user options	253
Setting Notification Options	253
Setting display options	254
Setting Syntax Display Options	255
Setting PMML Export Options	255
Setting User Information	256
Setting the mode	256
Customizing the Nodes Palette	259

Customizing the Palette Manager	259
Creating a Palette Tab	260
Displaying Palette Tabs on the Nodes Palette	260
Displaying Subpalettes on a Palette Tab	261
Creating a Subpalette	261
Changing a Palette Tab View	262
Performance considerations for streams and nodes	262
Order of Nodes	262
Node Caches	263
Performance: Process Nodes	264
Performance: Modeling Nodes	264
Performance: CLEM Expressions	265
Accessibility in IBM SPSS Modeler	265
Overview of Accessibility in IBM SPSS Modeler	265
Types of Accessibility Support	265
Accessibility for the Visually Impaired	266
Accessibility for Blind Users	267
Keyboard Accessibility	267
Shortcuts for navigating the main window	268
Shortcuts for Dialog Boxes and Tables	269
Shortcuts for Comments	270
Shortcuts for Cluster Viewer and Model Viewer	271
Shortcut Keys Example: Building Streams	272
Shortcut Keys Example: Editing Nodes	272
Using a Screen Reader	273
Using a Screen Reader with HTML Output	273
Accessibility in the Interactive Tree Window	274
Tips for use	274
Interference with Other Software	275
JAWS and Java	275
Using Graphs in IBM SPSS Modeler	275
Unicode support	275
Unicode Support in IBM SPSS Modeler	275
Batch User's Guide	
Batch Mode Execution	276
Introduction to Batch Mode	276
Working in Batch Mode	276
Invoking the Software	277
Using Command Line Arguments	277
Batch mode log files	278
Scripting in Batch Mode	278
Using Parameters in Batch Mode	279
Working with Output in Batch Mode	279
Source, Process, and Output Nodes	
Source Nodes	280
Overview	280
Setting Field Storage and Formatting	281
List storage and associated measurement levels	283
Unsupported control characters	283
Analytic Server Source node	284
Selecting a data source	284
Amending credentials	284
Supported nodes	284
Database source node	287
Setting Database Node Options	288
Adding a database connection	289
Potential database issues	290
Specifying preset values for a database connection	290
Settings for SQL Server	291
Settings for Oracle	291
Settings for IBM Db2 for z/OS, IBM Db2 LUW, and Teradata	291
Selecting a Database Table	292

Querying the database	293
Using Stream Parameters in an SQL Query	293
Using a custom database configuration file	294
Variable File Node	295
Setting options for the Variable File Node	296
Importing geospatial data into the Variable File Node	297
Fixed File Node	298
Setting options for the Fixed File node	298
Statistics File Node	299
Data Collection Node	299
Data Collection Import File Options	300
Data Collection Import Metadata Properties	301
Database Connection String	302
Advanced Properties	302
Importing Multiple Response Sets	302
Data Collection Column Import Notes	303
IBM Cognos source node	303
Cognos object icons	304
Importing Cognos data	304
Importing Cognos reports	305
Cognos connections	305
Cognos location selection	306
Specifying parameters for data or reports	306
IBM Cognos TM1 Source Node	306
Importing IBM Cognos TM1 data	306
TWC source node	307
SAS Source Node	308
Setting Options for the SAS Source Node	308
Selecting a Member	309
Excel Source node	309
XML Source Node	309
Selecting from Multiple Root Elements	310
Removing Unwanted Spaces from XML Source Data	310
User Input Node	311
Setting Options for the User Input Node	311
Simulation Generate Node	314
Setting Options for the Simulation Generate Node	315
Clone Field	318
Fit Details	318
Specify Parameters	319
Iterations	320
Distributions	320
Extension Import node	322
Extension Import node - Syntax tab	322
Extension Import node - Console Output tab	322
Filtering or renaming fields	323
Viewing and setting information about types	323
Geospatial Source Node	323
Setting Options for the Geospatial Source Node	323
JSON Source node	324
Common Source Node Tabs	324
Setting Measurement Levels in the Source Node	325
When to instantiate at the source node	325
Filtering Fields from the Source Node	326
Record Operations Nodes	326
Overview of Record Operations	326
Select Node	327
Sample Node	328
Sample node options	328
Cluster and Stratify Settings	330
Sample Sizes for Strata	330
Balance Node	331

Setting Options for the Balance Node	331
Aggregate Node	332
Setting options for the Aggregate node	333
Aggregate optimization settings	334
RFM Aggregate Node	334
Setting Options for the RFM Aggregate Node	335
Sort Node	335
Sort Optimization Settings	336
Merge Node	336
Types of Joins	337
Specifying a Merge Method and Keys	338
Selecting Data for Partial Joins	338
Specifying Conditions for a Merge	339
Specifying Ranked Conditions for a Merge	339
Filtering Fields from the Merge Node	340
Setting Input Order and Tagging	341
Merge Optimization Settings	341
Append Node	342
Setting Append Options	342
Distinct node	343
Distinct Optimization Settings	344
Distinct Composite Settings	344
Distinct Composite - Custom Tab	345
Streaming Time Series node	346
Streaming Time Series node - field options	346
Streaming Time Series node - data specification options	346
Streaming Time Series node - observations	347
Streaming Time Series node - time interval for analysis	347
Streaming Time Series node - aggregation and distribution options	348
Streaming Time Series node - missing value options	348
Streaming Time Series node - estimation period	349
Streaming Time Series node - build options	349
Streaming Time Series node - general build options	349
Transfer and transformation functions	351
Streaming Time Series node - model options	352
SMOTE node	353
SMOTE node Settings	353
Extension Transform node	354
Extension Transform node - Syntax tab	354
Extension Transform node - Console Output tab	355
Space-Time-Boxes Node	355
Defining Space-Time-Box density	357
Streaming TCM Node	357
Streaming TCM Node - Time Series options	357
Streaming TCM Node - Select Dimension Values	358
Streaming TCM Node - Observations options	358
Streaming TCM Node - Time Interval options	359
Streaming TCM Node - Aggregation and Distribution options	359
Streaming TCM Node - Missing Value options	359
Streaming TCM Node - General Data options	360
Streaming TCM Node - General Build options	360
Streaming TCM Node - Estimation Period options	360
Streaming TCM Node - Model options	361
CPLEX Optimization node	361
Setting options for the CPLEX Optimization node	362
Field Operations Nodes	362
Field Operations Overview	363
Automated Data Preparation	364
Fields tab (automated data preparation)	365
Settings tab (automated data preparation)	366
Field settings	366
Prepare dates & times (automated data preparation)	366

Excluding fields (automated data preparation)	367
Preparing inputs and targets	367
Construction and feature selection	368
Field names (automated data preparation)	368
Analysis tab (automated data preparation)	369
Field Processing Summary (automated data preparation)	369
Fields (automated data preparation)	370
Action Summary (automated data preparation)	371
Predictive Power (automated data preparation)	371
Fields Table (automated data preparation)	371
Field Details (automated data preparation)	371
Action Details (automated data preparation)	372
Generating a Derive node	374
Type Node	374
Viewing and setting information about types	323
Measurement levels	376
Geospatial measurement sublevels	377
Converting Continuous Data	378
What is instantiation?	378
Data Values	379
Using the Values Dialog Box	380
Specifying Values and Labels for Continuous Data	381
Values and Labels Subdialog Box	381
Specifying Values and Labels for Nominal and Ordinal Data	381
Specifying Values for a Flag	382
Specifying Values for Collection Data	382
Specifying values for geospatial data	382
Defining Missing Values	382
Checking Type Values	383
Setting the field role	383
Copying Type Attributes	384
Field Format Settings Tab	384
Setting Field Format options	385
Filtering or renaming fields	385
Setting filtering options	385
Truncating Field Names	386
Anonymizing Field Names	387
Editing Multiple Response Sets	387
Derive node	387
Setting Basic Options for the Derive Node	388
Deriving Multiple Fields	389
Selecting Multiple Fields	389
Setting Derive Formula Options	390
Setting derived list values	390
Deriving a list or geospatial field	391
Setting Derive Flag Options	391
Setting Derive Nominal Options	392
Setting Derive State Options	392
Setting Derive Count Options	393
Setting Derive Conditional Options	393
Recoding Values with the Derive Node	393
Filler node	393
Storage Conversion Using the Filler Node	394
Reclassify Node	395
Setting Options for the Reclassify Node	395
Reclassifying Multiple Fields	396
Storage and Measurement Level for Reclassified Fields	396
Anonymize Node	397
Setting Options for the Anonymize Node	397
Specifying How Field Values Will Be Anonymized	398
Anonymizing Field Values	398
Binning Node	398

Setting Options for the Binning Node	399
Fixed-Width Bins	400
Tiles (Equal Count or Sum)	400
Rank Cases	401
Mean/Standard Deviation	402
Optimal Binning	402
Cut Point Settings	403
Previewing the Generated Bins	403
RFM Analysis Node	404
RFM Analysis Node Settings	404
RFM Analysis Node Binning	405
Ensemble Node	405
Ensemble Node Settings	406
Partition Node	406
Partition Node Options	407
Set to Flag Node	408
Setting Options for the Set to Flag Node	408
Restructure Node	408
Setting Options for the Restructure Node	409
Transpose Node	410
Setting options for the Transpose node	410
History Node	411
Setting Options for the History Node	411
Field Reorder Node	412
Setting Field Reorder Options	412
Time Intervals Node	413
Time Interval - field options	413
Time Interval - build options	414
Reprojection Node	414
Setting options for the Reproject node	415
Graph Nodes	415
Common Graph Nodes Features	415
Aesthetics, Overlays, Panels, and Animation	416
Using the Output Tab	417
Using the Annotations Tab	418
3-D Graphs	418
Graphboard node	419
Graphboard Basic Tab	419
Field (Variable) Types	420
Graphboard Detailed Tab	421
Selecting map files for map visualizations	422
Available Built-in Graphboard Visualization Types	423
Creating Map Visualizations	426
Graphboard Examples	426
Example: Bar Chart with a Summary Statistic	427
Example: Stacked Bar Chart with a Summary Statistic	428
Example: Paneled Histogram	428
Example: Paneled Dot Plot	429
Example: Boxplot	430
Example: Pie Chart	431
Example: Heat Map	432
Example: Scatterplot Matrix (SPLOM)	433
Example: Choropleth (Color Map) of Sums	434
Example: Bar Charts on a Map	435
Graphboard appearance tab	436
Setting the Location of Templates, Stylesheets, and Maps	436
Using the IBM SPSS Collaboration and Deployment Services Repository as the Template, Stylesheet, and Map File Location	437
Managing Templates, Stylesheets, and Map Files	437
Converting and Distributing Map Shapefiles	438
Key Concepts for Maps	438
Using the Map Conversion Utility	439
Step 1 - Choose Destination and Source Files	439

Step 2 - Choose Map Key	439
Step 3 - Edit the Map	440
Smooth the map	440
Edit the feature labels	441
Compare to an External Data Source dialog box	441
Merge features	442
Name the Merged Feature dialog box	442
Move features	443
Delete features	443
Delete individual elements	443
Set the projection	444
Step 4 - Finish	444
Distributing map files	445
Plot Node	445
Plot Node Tab	447
Plot Options Tab	448
Plot Appearance Tab	449
Using a Plot Graph	450
Multiplot Node	450
Multiplot Plot Tab	451
Multiplot Appearance Tab	452
Using a Multiplot Graph	453
Time Plot Node	453
Time Plot Tab	454
Time Plot Appearance Tab	455
Using a Time Plot Graph	456
Distribution Node	456
Distribution Plot Tab	457
Distribution Appearance Tab	457
Using a Distribution Node	458
Histogram Node	459
Histogram Plot Tab	460
Histogram Options Tab	460
Histogram Appearance Tab	461
Using Histograms	461
Collection Node	462
Collection Plot Tab	463
Collection Options Tab	463
Collection Appearance Tab	463
Using a Collection Graph	464
Web Node	465
Web Plot Tab	466
Web Options Tab	467
Web Appearance Tab	468
Using a Web Graph	468
Adjusting Web Thresholds	470
Creating a web summary	471
Evaluation node	471
Evaluation Plot Tab	474
Evaluation Options tab	475
Evaluation Appearance Tab	476
Reading the Results of a Model Evaluation	476
Using an Evaluation Chart	477
Map Visualization Node	478
Map Visualization Plot Tab	478
Change map layers	479
Map Visualization Appearance Tab	481
t-SNE node	481
t-SNE node Expert options	482
t-SNE node Output options	483
Accessing and plotting t-SNE data	483
t-SNE model nuggets	484

E-Plot (Beta) node	484
E-Plot (Beta) node Plot tab	484
E-Plot (Beta) node Options tab	485
E-Plot (Beta) Appearance tab	485
Using an e-plot graph	485
Working with Graph Output	487
Exploring Graphs	487
Using Bands	488
Using Regions	490
Using Marked Elements	492
Generating Nodes from Graphs	493
Editing Visualizations	495
General Rules for Editing Visualizations	496
Editing and Formatting Text	497
Changing Colors, Patterns, Dashings, and Transparency	497
Rotating and Changing the Shape and Aspect Ratio of Point Elements	498
Changing the size of graphic elements	498
Specifying Margins and Padding	498
Formatting Numbers	499
Changing the Axis and Scale Settings	499
Editing Categories	500
Changing the Orientation Panels	501
Transforming the Coordinate System	502
Changing Statistics and Graphic Elements	502
Changing the Position of the Legend	503
Copying a Visualization and Visualization Data	503
Graphboard Editor Keyboard Shortcuts	504
Adding Titles and Footnotes	504
Using Graph Stylesheets	505
Applying stylesheets	505
Printing, saving, copying, and exporting graphs	506
Setting Header and Footer Preferences	508
Output Nodes	508
Overview of Output Nodes	508
Managing output	509
Viewing output	509
Publish to Web	510
Publishing Output to the Web	510
Viewing Published Output Over the Web	511
Viewing Output in an HTML Browser	511
Exporting Output	511
Selecting cells and columns	512
Table node	512
Table Node Settings Tab	512
Output Node Output Tab	512
Table Browser	513
Matrix node	514
Matrix Node Settings Tab	514
Matrix Node Appearance Tab	515
Matrix node output browser	515
Analysis Node	516
Analysis Node Analysis Tab	516
Analysis Output Browser	517
Data Audit node	518
Data Audit Node Settings Tab	519
Data Audit Quality Tab	520
Data Audit Output Browser	520
Viewing and Generating Graphs	521
Display Statistics	521
Data Audit Browser Quality Tab	522
Imputing Missing Values	523
Handling Outliers and Extreme Values	524

Filtering Fields with Missing Data	524
Selecting Records with Missing Data	525
Generating Other Nodes for Data Preparation	525
Transform Node	526
Transform Node Options Tab	526
Transform Node Output Tab	527
Transform Node Output Viewer	527
Generating Nodes for the Transformations	527
Generating Graphs	528
Other Operations	528
Statistics Node	528
Statistics Node Settings Tab	528
Correlation Settings	529
Statistics Output Browser	529
Generating a Filter Node from Statistics	530
Means Node	530
Comparing Means for Independent Groups	531
Comparing Means Between Paired Fields	531
Means Node Options	532
Means Node Output Browser	532
Means Output Comparing Groups within a Field	532
Means Output Comparing Pairs of Fields	533
Report Node	533
Report Node Template Tab	534
Report Node Output Browser	535
Set Globals Node	535
Set Globals Node Settings Tab	535
Simulation Fitting Node	536
Distribution Fitting	536
Simulation Fitting Node Settings Tab	537
Simulation Evaluation Node	538
Simulation Evaluation Node Settings Tab	538
Simulation Evaluation node output	540
Navigation Panel	540
Chart output	541
Chart Options	542
Extension Output node	543
Extension Output node - Syntax tab	543
Extension Output node - Console Output tab	544
Extension Output node - Output tab	544
Extension Output Browser	544
Extension Output Browser - Text Output tab	544
Extension Output Browser - Graph Output tab	545
KDE nodes	545
KDE Modeling node and KDE Simulation node Fields	545
KDE nodes Build Options	545
KDE Modeling node and KDE Simulation node Model Options	546
IBM SPSS Statistics Helper Applications	546
Export Nodes	547
Overview of Export Nodes	547
Database Export Node	548
Database Node Export Tab	548
Database Export Merge Options	549
Database export schema options	550
Options for SQL Server	551
Options for Oracle	551
Database Export Index Options	552
Database export advanced options	553
Bulk loader programming	554
Bulk loading data to IBM Db2 databases	555
Bulk loading data to IBM Netezza databases	555
Bulk loading data to Oracle databases	556

Bulk loading data to SQL Server databases	556
Bulk loading data to Teradata databases	557
Developing bulk loader programs	557
Testing bulk loader programs	558
Flat File Export Node	559
Flat File Export Tab	559
Statistics Export Node	559
Statistics Export Node - Export Tab	560
Renaming or Filtering Fields for IBM SPSS Statistics	560
Data Collection Export Node	561
Analytic Server Export node	561
IBM Cognos export node	562
Cognos connection	562
ODBC connection	562
IBM Cognos TM1 Export Node	563
Connecting to an IBM Cognos TM1 cube to export data	564
Mapping IBM Cognos TM1 data for export	564
SAS Export Node	565
SAS Export Node Export Tab	565
Excel Export Node	565
Excel Node Export Tab	565
Extension Export node	566
Extension Export node - Syntax tab	566
Extension Export node - Console Output tab	566
XML Export Node	567
Writing XML Data	567
XML Mapping Records Options	568
XML Mapping Fields Options	568
XML Mapping Preview	568
JSON Export node	568
Common Export node tabs	569
Publishing streams	569
IBM SPSS Statistics Nodes	570
IBM SPSS Statistics Nodes - Overview	570
Statistics File Node	299
Statistics Transform Node	571
Statistics Transform Node - Syntax Tab	572
Allowable Syntax	572
Statistics Model Node	573
Statistics Model Node - Model Tab	574
Statistics Model Node - Model Nugget Summary	574
Statistics Output Node	574
Statistics Output Node - Syntax Tab	575
Statistics Output Node - Output Tab	576
Statistics Export Node	559
Statistics Export Node - Export Tab	560
Renaming or Filtering Fields for IBM SPSS Statistics	560
SuperNodes	578
Overview of SuperNodes	578
Types of SuperNodes	578
Source SuperNodes	579
Process SuperNodes	579
Terminal SuperNodes	579
Creating SuperNodes	579
Nesting SuperNodes	580
Locking SuperNodes	580
Locking and unlocking a SuperNode	581
Editing a locked SuperNode	581
Editing SuperNodes	581
Modifying SuperNode types	582
Annotating and renaming SuperNodes	582
SuperNode parameters	582

Defining SuperNode Parameters	582
Setting Values for SuperNode Parameters	583
Using SuperNode Parameters to Access Node Properties	583
SuperNodes and caching	584
SuperNodes and scripting	584
Saving and loading SuperNodes	584
Modeling Nodes	
Modeling Overview	585
Overview of modeling nodes	585
Building Split Models	588
Splitting and Partitioning	588
Modeling nodes supporting split models	589
Features Affected by Splitting	589
Modeling Node Fields Options	590
Using Frequency and Weight Fields	591
Modeling Node Analyze Options	592
Propensity Scores	593
Misclassification Costs	593
Model Nuggets	594
Model Links	594
Defining and Removing Model Links	595
Copying and Pasting Model Links	595
Model Links and SuperNodes	596
Replacing a model	596
The models palette	597
Browsing model nuggets	597
Model Nugget Summary / Information	598
Predictor Importance	599
Filtering Variables Based on Importance	599
Ensemble Viewer	600
Models for ensembles	600
Model Summary (ensemble viewer)	600
Predictor Importance (ensemble viewer)	601
Predictor Frequency (ensemble viewer)	601
Component Model Accuracy (ensemble viewer)	601
Component Model Details (ensemble viewer)	601
Automatic Data Preparation (ensemble viewer)	601
Model Nuggets for Split Models	602
Split Model Viewer	602
Using Model Nuggets in Streams	602
Regenerating a modeling node	603
Importing and exporting models as PMML	603
Model types supporting PMML	604
Publishing models for a scoring adapter	605
Unrefined Models	605
Generated Statistical Models Advanced Output	605
Cluster Models Model Tab	606
Generated Rule Set/Decision Tree Model Tab	606
Screening Models	606
Screening Fields and Records	606
Feature Selection node	607
Feature Selection Model Settings	607
Feature Selection Options	608
Feature Selection Model Nuggets	609
Feature Selection Model Results	609
Selecting Fields by Importance	609
Generating a Filter from a Feature Selection Model	610
Anomaly Detection Node	610
Anomaly Detection Model Options	611
Anomaly Detection Expert Options	611
Anomaly Detection Model Nuggets	612
Anomaly Detection model details	612

Anomaly Detection Model Summary	613
Anomaly Detection Model Settings	613
Automated Modeling Nodes	613
Automated Modeling Node Algorithm Settings	614
Automated Modeling Node Stopping Rules	615
Execution Feedback	615
Auto Classifier node	615
Auto Classifier node model options	616
Auto Classifier Node Expert Options	617
Auto Classifier Node Discard Options	618
Auto Classifier node settings	619
Auto Numeric node	619
Auto Numeric node model options	620
Auto Numeric Node Expert Options	621
Auto Numeric node settings	622
Auto Cluster node	622
Auto Cluster Node Model Options	623
Auto Cluster Node Expert Options	623
Auto Cluster Node Discard Options	624
Automated Model Nuggets	624
Generating Nodes and Models	626
Generating Evaluation Charts	626
Evaluation Graphs	627
Automated Model Nugget Summary	627
Continuous machine learning	627
Decision Trees	634
Decision Tree Models	634
The Interactive Tree Builder	635
Growing and Pruning the Tree	636
Defining Custom Splits	637
Viewing Predictor Details	637
Split Details and Surrogates	638
Customizing the Tree View	638
Gains	639
Classification Gains	640
Classification Profits and ROI	641
Regression Gains	641
Gains Charts	642
Gains-Based Selection	642
Risks	643
Saving Tree Models and Results	643
Generating a Model from the Tree Builder	644
Tree-Growing Directives	644
Updating Tree Directives	646
Exporting Model, Gain, and Risk Information	646
Generating Filter and Select Nodes	647
Generating a Rule Set from a Decision Tree	647
Building a Tree Model Directly	648
Decision Tree Nodes	648
C&R Tree Node	649
CHAID Node	650
QUEST Node	650
Decision Tree Node Fields Options	650
Decision Tree Node Build Options	651
Decision Tree Nodes - Objectives	651
Decision Tree Nodes - Basics	652
Decision Tree Nodes - Stopping Rules	652
Decision Tree Nodes - Ensembles	652
C&R Tree and QUEST Nodes - Costs & Priors	653
CHAID Node - Costs	654
C&R Tree Node - Advanced	654
QUEST Node - Advanced	654

CHAID Node - Advanced	654
Decision Tree Node Model Options	655
C5.0 Node	656
C5.0 Node Model Options	656
Tree-AS node	657
Tree-AS node fields options	658
Tree-AS node build options	658
Tree-AS node - basics	658
Tree-AS node - growing	659
Tree-AS node - stopping rules	659
Tree-AS node - costs	660
Tree-AS node model options	660
Tree-AS model nugget	660
Tree-AS model nugget output	660
Tree-AS model nugget settings	661
Random Trees node	661
Random Trees node fields options	662
Random Trees node build options	663
Random Trees node - basics	663
Random Trees node - costs	663
Random Trees node - advanced	664
Random Trees node model options	664
Random Trees model nugget	664
Random Trees model nugget output	664
Random Trees model nugget settings	666
C&R Tree, CHAID, QUEST, and C5.0 decision tree model nuggets	666
Single Tree Model Nuggets	667
Decision Tree Model Rules	667
Decision Tree Model Viewer	669
Decision Tree/Rule Set model nugget settings	669
Boosted C5.0 Models	670
Generating Graphs	670
Model nuggets for boosting, bagging, and very large datasets	671
C&R Tree, CHAID, QUEST, C5.0, and Apriori rule set model nuggets	671
Rule Set Model Tab	672
Bayesian Network Models	673
Bayesian Network Node	673
Bayesian Network Node Model Options	674
Bayesian Network Node Expert Options	675
Bayesian Network Model Nuggets	676
Bayesian Network Model Settings	676
Bayesian Network Model Summary	677
Neural networks	677
The neural networks model	678
Using neural networks with legacy streams	679
Objectives (neural networks)	679
Basics (neural networks)	680
Stopping rules (neural networks)	681
Ensembles (neural networks)	681
Advanced (neural networks)	682
Model Options (neural networks)	683
Model Summary (neural networks)	683
Predictor Importance (neural networks)	684
Predicted By Observed (neural networks)	684
Classification (neural networks)	685
Network (neural networks)	686
Settings (neural networks)	686
Decision List	687
Decision List Model Options	688
Decision List Node Expert Options	689
Decision List Model Nugget	689
Decision List Model Nugget Settings	690

Decision List Viewer	690
Working Model Pane	691
Alternatives Tab	692
Snapshots Tab	692
Working with Decision List Viewer	693
Mining Tasks	693
Running Mining Tasks	694
Creating and Editing a Mining Task	694
New Settings	695
Edit Advanced Parameters	696
Customize Available Fields	696
Organizing Data Selections	696
Specify Selection Condition	697
Insert Value	697
Undo and Redo Actions	697
Segment Rules	698
Inserting Segments	698
Editing Segment Rules	699
Insert/Edit Condition	699
Deleting Segment Rule Conditions	699
Copying Segments	700
Alternative Models	700
Customizing a Model	701
Prioritizing Segments	701
Deleting Segments	701
Excluding Segments	702
Change Target Value	702
Generate New Model	702
Model Assessment	703
Organizing Model Measures	703
Refreshing Measures	704
Assessment in Excel	704
Choose Inputs for Custom Measures	705
MS Excel Integration Setup	705
Changing the Model Measures	705
Visualizing Models	706
Gains Chart	707
Chart Options	707
Statistical Models	707
Linear Node	708
Linear models	709
Objectives (linear models)	709
Basics (linear models)	710
Model selection (linear models)	710
Ensembles (linear models)	711
Advanced (linear models)	711
Model Options (linear models)	711
Model Summary (linear models)	711
Automatic Data Preparation (linear models)	712
Predictor Importance (linear models)	712
Predicted By Observed (linear models)	712
Residuals (linear models)	712
Outliers (linear models)	712
Effects (linear models)	713
Coefficients (linear models)	713
Estimated means (linear models)	713
Model Building Summary (linear models)	713
Settings (linear models)	714
Linear-AS Node	714
Linear-AS models	714
Basics (linear-AS models)	715
Model Selection (linear-AS models)	715

Model Options (linear-AS models)	716
Interactive Output (linear-AS models)	716
Settings (linear-AS models)	716
Logistic Node	717
Logistic Node Model Options	717
Adding Terms to a Logistic Regression Model	720
Logistic Node Expert Options	720
Logistic Regression Convergence Options	721
Logistic Regression Advanced Output	721
Logistic Regression Stepping Options	722
Logistic Model Nugget	723
Logistic Nugget Model Details	723
Logistic Model Nugget Summary	724
Logistic Model Nugget Settings	724
Logistic Model Nugget Advanced Output	725
PCA/Factor Node	726
PCA/Factor Node Model Options	726
PCA/Factor Node Expert Options	727
PCA/Factor Node Rotation Options	727
PCA/Factor Model Nugget	728
PCA/Factor Model Nugget Equations	728
PCA/Factor Model Nugget Summary	728
PCA/Factor Model Nugget Advanced Output	729
Discriminant node	729
Discriminant Node Model Options	730
Discriminant Node Expert Options	730
Discriminant Node Output Options	730
Discriminant Node Stepping Options	731
Discriminant Model Nugget	732
Discriminant Model Nugget Advanced Output	732
Discriminant Model Nugget Settings	733
Discriminant Model Nugget Summary	733
GenLin Node	733
GenLin Node Field Options	734
GenLin Node Model Options	734
GenLin Node Expert Options	735
Generalized Linear Models Iterations	737
Generalized Linear Models Advanced Output	737
GenLin Model Nugget	738
GenLin Model Nugget Advanced Output	739
GenLin Model Nugget Settings	739
GenLin Model Nugget Summary	740
Generalized Linear Mixed Models	740
GLMM Node	740
Generalized linear mixed models	740
Target (generalized linear mixed models)	741
Fixed Effects (generalized linear mixed models)	743
Add a Custom Term (generalized linear mixed models)	744
Random Effects (generalized linear mixed models)	745
Random Effect Block (generalized linear mixed models)	745
Weight and Offset (generalized linear mixed models)	746
General Build Options (generalized linear mixed models)	747
Estimation (generalized linear mixed models)	748
General (generalized linear mixed models)	749
Estimated Means (generalized linear mixed models)	749
Model view (generalized linear mixed models)	751
Model Summary (generalized linear mixed models)	751
Data Structure (generalized linear mixed models)	751
Predicted by Observed (generalized linear mixed models)	752
Classification (generalized linear mixed models)	752
Fixed Effects (generalized linear mixed models)	753
Fixed Coefficients (generalized linear mixed models)	754

Random Effect Covariances (generalized linear mixed models)	754
Covariance Parameters (generalized linear mixed models)	755
Estimated Means: Significant Effects (generalized linear mixed models)	756
Estimated Means: Custom Effects (generalized linear mixed models)	756
Settings (generalized linear mixed models)	757
GLE Node	758
Target (GLE models)	758
Model effects (GLE models)	760
Add a custom term (GLE models)	760
Weight and Offset (GLE models)	761
Build options (GLE models)	761
Estimation (GLE models)	761
Model selection (GLE models)	762
Model options (GLE models)	763
GLE model nugget	763
GLE model nugget output	763
GLE model nugget settings	764
Cox Node	764
Cox Node Fields Options	765
Cox Node Model Options	765
Adding Terms to a Cox Regression Model	766
Cox Node Expert Options	766
Cox Node Convergence Criteria	766
Cox Node Advanced Output Options	767
Cox Node Stepping Criteria	767
Cox Node Settings Options	768
Cox Model Nugget	768
Cox Regression Output Settings	769
Cox Regression Advanced Output	769
Clustering models	769
Kohonen node	770
Kohonen Node Model Options	771
Kohonen Node Expert Options	771
Kohonen Model Nuggets	772
Kohonen Model Summary	772
K-Means Node	773
K-Means Node Model Options	773
K-Means Node Expert Options	774
K-Means Model Nuggets	774
K-Means Model Summary	775
TwoStep Cluster node	775
TwoStep Cluster Node Model Options	775
TwoStep Cluster Model Nuggets	776
TwoStep Model Summary	776
TwoStep-AS Cluster node	777
Twostep-AS cluster analysis	777
Fields tab (Twostep-AS Cluster)	777
Basics (Twostep-AS Cluster)	777
Feature Tree Criteria (Twostep-AS Cluster)	778
Standardize	779
Feature Selection	779
Model Output	780
Model Options	781
TwoStep-AS Cluster Model Nuggets	781
TwoStep-AS Cluster Model Nugget Settings	781
K-Means-AS node	781
K-Means-AS node Fields	782
K-Means-AS node Build Options	782
The Cluster Viewer	782
Cluster Viewer - Model Tab	783
Model Summary View	784
Clusters View	784

Transpose Clusters and Features	785
Sort Features	785
Sort Clusters	786
Cell Contents	786
Cluster Predictor Importance View	787
Cluster Sizes View	787
Cell Distribution View	787
Cluster Comparison View	788
Navigating the Cluster Viewer	788
Generating Graphs from Cluster Models	790
Association Rules	790
Tabular versus Transactional Data	792
Apriori node	792
Apriori Node Model Options	792
Apriori Node Expert Options	793
CARMA Node	794
CARMA Node Fields Options	794
CARMA Node Model Options	795
CARMA Node Expert Options	795
Association Rule Model Nuggets	796
Association Rule model nugget details	797
Specifying Filters for Rules	798
Generating Graphs for Rules	799
Association Rule Model Nugget Settings	799
Association Rule Model Nugget Summary	800
Generating a Rule Set from an Association Model Nugget	801
Generating a Filtered Model	801
Scoring Association Rules	802
Deploying Association Models	803
Sequence node	804
Sequence Node Fields Options	804
Sequence Node Model Options	805
Sequence Node Expert Options	805
Sequence Model Nuggets	806
Sequence Model Nugget Details	807
Sequence Model Nugget Settings	809
Sequence Model Nugget Summary	809
Generating a Rule SuperNode from a Sequence Model Nugget	809
Association Rules node	810
Association Rules - Fields Options	811
Association Rules - Rule building	811
Association Rules - Transformations	812
Association Rules - Output	813
Association Rules - Model Options	814
Association Rules Model Nuggets	814
Association Rules Model Nugget Details	815
Association Rules Model Nugget Settings	815
Time Series Models	816
Why forecast?	816
Time series data	817
Characteristics of time series	817
Trends	817
Seasonal cycles	818
Nonseasonal cycles	818
Pulses and steps	818
Outliers	819
Autocorrelation and partial autocorrelation functions	820
Series transformations	821
Predictor series	821
Spatio-Temporal Prediction modeling node	822
Spatio-Temporal Prediction - Fields Options	822
Spatio-Temporal Prediction - Time Intervals	823

Spatio-Temporal Prediction - Basic Build Options	824
Spatio-Temporal Prediction - Advanced Build Options	824
Spatio-Temporal Prediction - Output	825
Spatio-Temporal Prediction - Model Options	826
Spatio-Temporal Prediction Model Nugget	826
Spatio-Temporal Prediction Model Settings	826
TCM Node	826
Temporal Causal Models	827
Time Series to Model (Temporal Causal Modeling)	827
Select Dimension Values (Temporal Causal Modeling)	828
Observations (Temporal Causal Modeling)	828
Time Interval for Analysis (Temporal Causal Modeling)	829
Aggregation and Distribution (Temporal Causal Modeling)	829
Missing Values (Temporal Causal Modeling)	830
General Data Options (Temporal Causal Modeling)	830
General Build Options (Temporal Causal Modeling)	831
Series to Display (Temporal Causal Modeling)	831
Output Options (Temporal Causal Modeling)	832
Estimation Period (Temporal Causal Modeling)	833
Model Options (Temporal Causal Modeling)	833
Interactive Output (Temporal Causal Modeling)	834
TCM Model Nugget	834
TCM Model Nugget Settings	835
Temporal Causal Model Scenarios	835
Defining the Scenario Period (Temporal Causal Model Scenarios)	835
Adding Scenarios and Scenario Groups (Temporal Causal Model Scenarios)	836
Scenario Definition (Temporal Causal Model Scenarios)	837
Select Dimension Values (Temporal Causal Model Scenarios)	837
Scenario Group Definition (Temporal Causal Model Scenarios)	838
Options (Temporal Causal Model Scenarios)	839
Time Series node	839
Time Series node - field options	840
Time Series node - data specification options	840
Time Series node - observations	840
Time Series node - time interval for analysis	841
Time Series node - aggregation and distribution options	841
Time Series node - missing value options	842
Time Series node - estimation period	842
Time Series node - build options	842
Time Series node - general build options	843
Transfer and transformation functions	845
Time Series node - build output options	846
Time Series node - model options	846
Time Series model nugget	847
Time Series model nugget output	847
Time Series model nugget settings	848
Self-Learning Response Node Models	849
SLRM node	849
SLRM Node Fields Options	850
SLRM Node Model Options	850
SLRM Node Settings Options	851
SLRM Model Nuggets	852
SLRM Model Settings	852
Support Vector Machine Models	853
About SVM	853
How SVM Works	853
Tuning an SVM Model	854
SVM node	855
SVM Node Model Options	855
SVM Node Expert Options	855
SVM Model Nugget	856
SVM Model Settings	857

LSVM Node	857
LSVM Node Model Options	858
LSVM Build Options	858
LSVM Model Nugget (interactive output)	858
LSVM Model Settings	859
Nearest Neighbor Models	859
KNN node	860
KNN Node Objectives Options	860
KNN Node Settings	861
Model	861
Neighbors	862
Feature Selection	862
Cross-Validation	863
Analyze	863
KNN Model Nugget	864
Nearest Neighbor Model View	864
Model View (Nearest Neighbor Analysis)	865
Predictor Space (Nearest Neighbor Analysis)	865
Changing the axes on the Predictor Space chart	865
Predictor Importance (Nearest Neighbor Analysis)	866
Nearest Neighbor Distances (Nearest Neighbor Analysis)	866
Peers (Nearest Neighbor Analysis)	866
Quadrant Map (Nearest Neighbor Analysis)	866
Predictor selection error log (Nearest Neighbor Analysis)	866
Classification Table (Nearest Neighbor Analysis)	867
Error Summary (Nearest Neighbor Analysis)	867
KNN Model Settings	867
Glossary	867
Python nodes	872
SMOTE node	353
SMOTE node Settings	353
XGBoost Linear node	873
XGBoost Linear node Fields	874
XGBoost Linear node Build Options	874
XGBoost Linear node Model Options	875
XGBoost Tree node	875
XGBoost Tree node Fields	875
XGBoost Tree node Build Options	875
XGBoost Tree node Model Options	877
t-SNE node	481
t-SNE node Expert options	482
t-SNE node Output options	483
t-SNE model nuggets	484
Gaussian Mixture node	879
Gaussian Mixture node Fields	880
Gaussian Mixture node Build Options	880
Gaussian Mixture node Model Options	881
KDE nodes	545
KDE Modeling node and KDE Simulation node Fields	545
KDE nodes Build Options	545
KDE Modeling node and KDE Simulation node Model Options	546
Random Forest node	883
Random Forest node Fields	883
Random Forest node Build Options	883
Random Forest node Model Options	884
Random Forest model nuggets	884
HDBSCAN node	885
HDBSCAN node Fields	885
HDBSCAN node Build Options	885
HDBSCAN node Model Options	886
One-Class SVM node	887
One-Class SVM node Fields	887
One-Class SVM node Expert	887

One-Class SVM node Options	888
Spark nodes	888
Isotonic-AS node	889
Isotonic-AS node Fields	889
Isotonic-AS node Build Options	889
Isotonic-AS model nuggets	889
XGBoost-AS node	889
XGBoost-AS node Fields	890
XGBoost-AS node Build Options	890
XGBoost-AS node Model Options	892
K-Means-AS node	781
K-Means-AS node Fields	782
K-Means-AS node Build Options	782
MultiLayerPerceptron-AS node	893
MultiLayerPerceptron-AS node Fields	893
MultiLayerPerceptron-AS node Build Options	894
MultiLayerPerceptron node Model Options	894
Scripting and Automation	
Scripting and the Scripting Language	894
Scripting overview	895
Types of Scripts	895
Stream Scripts	895
Stream script example: Training a neural net	896
Jython code size limits	897
Standalone Scripts	897
Standalone script example: Saving and loading a model	898
Standalone script example: Generating a Feature Selection model	898
SuperNode Scripts	899
SuperNode Script Example	899
Looping and conditional execution in streams	900
Looping in streams	901
Creating an iteration key for looping in streams	901
Creating an iteration variable for looping in streams	902
Selecting fields for iterations	903
Conditional execution in streams	903
Executing and interrupting scripts	904
The Scripting Language	905
Scripting language overview	905
Python and Jython	905
Python Scripting	906
Operations	906
Lists	907
Strings	908
Remarks	909
Statement Syntax	909
Identifiers	910
Blocks of Code	910
Passing Arguments to a Script	911
Examples	911
Mathematical Methods	912
Using Non-ASCII characters	913
Object-Oriented Programming	914
Defining a Class	914
Creating a Class Instance	915
Adding Attributes to a Class Instance	915
Defining Class Attributes and Methods	916
Hidden Variables	916
Inheritance	917
Scripting in IBM SPSS Modeler	917
Types of scripts	917
Streams, SuperNode streams, and diagrams	918
Streams	918

SuperNode streams	918
Diagrams	919
Executing a stream	919
The scripting context	919
Referencing existing nodes	920
Finding nodes	920
Setting properties	921
Creating nodes and modifying streams	921
Creating nodes	922
Linking and unlinking nodes	922
Importing, replacing, and deleting nodes	923
Traversing through nodes in a stream	923
Clearing, or removing, items	924
Getting information about nodes	924
The Scripting API	925
Introduction to the Scripting API	925
Example 1: searching for nodes using a custom filter	926
Example 2: allowing users to obtain directory or file information based on their privileges	926
Metadata: Information about data	926
Accessing Generated Objects	928
Handling errors	929
Stream, Session, and SuperNode Parameters	930
Global Values	933
Working with Multiple Streams: Standalone Scripts	933
Scripting tips	934
Modifying stream execution	934
Looping through nodes	934
Accessing Objects in the IBM SPSS Collaboration and Deployment Services Repository	935
Generating an encoded password	937
Script checking	937
Scripting from the command line	937
Compatibility with previous releases	937
Accessing stream execution results	937
Table content model	938
XML Content Model	939
JSON Content Model	941
Column statistics content model and pairwise statistics content model	941
Command Line Arguments	943
Invoking the software	943
Using command line arguments	944
System arguments	944
Parameter arguments	945
Server connection arguments	946
IBM SPSS Collaboration and Deployment Services Repository Connection Arguments	947
IBM SPSS Analytic Server connection arguments	947
Combining Multiple Arguments	947
Properties Reference	948
Properties reference overview	948
Syntax for properties	948
Structured properties	949
Abbreviations	949
Node and stream property examples	950
Node properties overview	950
Common Node Properties	950
Stream properties	951
Source Node Properties	953
Source node common properties	953
asimport Properties	956
cognosimport Node Properties	957
databasename Properties	958
datacollectionimportnode Properties	958
excelimportnode Properties	960

extensionimportnode properties	960
fixedfilenode Properties	962
gsdata_import Node Properties	963
jsonimportnode Properties	964
sasimportnode Properties	964
simgennode properties	965
statisticsimportnode Properties	966
tm1odataimport Node Properties	966
tm1import Node Properties (deprecated)	967
twcimport node properties	968
userinputnode properties	968
variablefilenode Properties	968
xmlexportnode Properties	970
Record Operations Node Properties	971
appendnode properties	971
aggregatenode properties	971
balancenode properties	972
cplexoptnode properties	972
derive_stbnode properties	973
distinctnode properties	975
extensionprocessnode properties	976
mergenode properties	977
rffmaggrenode properties	978
samplenode properties	979
selectnode properties	980
sortnode properties	980
spacetimeboxes properties	981
streamingtimeseries Properties	982
Field Operations Node Properties	984
anonymizenode properties	985
autodataprepnode properties	985
astimeintervalsnode properties	988
binningnode properties	988
derivenode properties	990
ensemblenode properties	991
fillernode properties	992
filternode properties	992
historynode properties	993
partitionnode properties	993
reclassifynode properties	994
reordernode properties	995
reprojectnode properties	995
restructurenode properties	995
rfmanalysisnode properties	996
settoflagnode properties	996
statisticstransformnode properties	997
timeintervalsnode properties (deprecated)	997
transposenode properties	1000
typenode properties	1000
Graph Node Properties	1004
Graph node common properties	1004
collectionnode Properties	1004
distributionnode Properties	1005
evaluationnode Properties	1006
graphboardnode Properties	1007
histogramnode Properties	1008
mapvisualization properties	1009
multiplotnode Properties	1011
plotnode Properties	1012
timeplotnode Properties	1014
eplotnode Properties	1015
tsnenode Properties	1015

webnode Properties	1016
Modeling Node Properties	1017
Common modeling node properties	1018
anomalydetectionnode properties	1018
apriorinode properties	1019
associationrulesnode properties	1020
autoclassifiernode properties	1021
Setting Algorithm Properties	1022
autoclusternode properties	1023
autonumericnode properties	1024
bayesnetnode properties	1025
c50node properties	1026
carmanode properties	1027
cartnode properties	1028
chaidnode properties	1029
coxregnodel properties	1031
decisionlistnode properties	1032
discriminantnode properties	1032
extensionmodelnode properties	1034
factornode properties	1036
featureselectionnode properties	1037
genlinnode properties	1038
glmmnode properties	1040
gle properties	1042
kmeansnode properties	1044
kmeansasnode properties	1045
knnnode properties	1045
kohonennnode properties	1046
linearnode properties	1047
linearasnode properties	1048
logregnodel properties	1048
lsvmnode properties	1051
neuralnetnode properties	1052
neuralnetworknode properties	1053
questnode properties	1054
randomtrees properties	1056
regressionnode properties	1056
sequencenode properties	1058
slrnmnode properties	1059
statisticsmodelnode properties	1059
stpmnode properties	1060
svmnnode properties	1061
tcmnode Properties	1062
ts properties	1064
treeas properties	1067
twostepnode Properties	1068
twostepAS Properties	1069
Model nugget node properties	1070
applyanomalydetectionnode Properties	1070
applyapriorinode Properties	1071
applyassociationrulesnode Properties	1072
applyautoclassifiernode Properties	1073
applyautoclusternode Properties	1074
applyautonumericnode Properties	1074
applybayesnetnode Properties	1075
applyc50node Properties	1076
applycarmanode Properties	1076
applycartnode Properties	1077
applychaidnode Properties	1077
applycoxregnodel Properties	1078
applydecisionlistnode Properties	1079
applydiscriminantnode Properties	1080

applyextension properties	1080
applyfactornode Properties	1082
applyfeatureselectionnode Properties	1082
applygeneralizedlinearnode Properties	1083
applyglmnode Properties	1084
applygle Properties	1085
applygmm properties	1085
applykmeansnode Properties	1086
applyknnnode Properties	1086
applykohonennode Properties	1087
applylinearnode Properties	1088
applylinearasnode Properties	1088
applylogregnode Properties	1089
applylsvmnode Properties	1090
applyneuralnetnode Properties	1091
applyneuralnetworknode properties	1091
applyocsvmnode properties	1092
applyquestnode Properties	1092
applyrandomtrees Properties	1093
applyregressionnode Properties	1094
applyselflearningnode properties	1094
applysequencenode Properties	1094
applysvmnode Properties	1095
applystpnode Properties	1096
applytcnnode Properties	1097
applyts Properties	1097
applytimeseriesnode Properties (deprecated)	1098
applytreeas Properties	1099
applytwostepnode Properties	1100
applytwostepAS Properties	1100
applyxgboosttreenode properties	1101
applyxgboostlinearnode properties	1101
hdbscan nugget properties	1101
kdeapply properties	1101
Database modeling node properties	1102
Node Properties for Microsoft Modeling	1102
Microsoft Modeling Node Properties	1102
Algorithm Parameters	1103
Microsoft Model Nugget Properties	1104
Node Properties for Oracle Modeling	1105
Oracle Modeling Node Properties	1105
Oracle Model Nugget Properties	1109
Node Properties for IBM Netezza Analytics Modeling	1110
Netezza Modeling Node Properties	1110
Netezza Model Nugget Properties	1115
Output node properties	1115
analysisnode properties	1116
dataauditnode properties	1116
extensionoutputnode properties	1117
kdeexport properties	1118
matrixnode properties	1119
meansnode properties	1119
reportnode properties	1120
setglobalsnode properties	1121
simevalnode properties	1121
simfitnode properties	1122
statisticsnode properties	1122
statisticsoutputnode Properties	1123
tablenode properties	1123
transformnode properties	1124
Export Node Properties	1125
Common Export Node Properties	1125

aselexport Properties	1126
cognosexportnode Properties	1126
databaseexportnode properties	1127
datacollectionexportnode Properties	1129
excelexportnode Properties	1130
extensionexportnode properties	1130
jsonexportnode Properties	1131
outputfilenode Properties	1131
sasexportnode Properties	1132
statisticsexportnode Properties	1133
tm1odataexport Node Properties	1133
tm1export Node Properties (deprecated)	1134
xmlexportnode Properties	1135
IBM SPSS Statistics Node Properties	1136
statisticsimportnode Properties	1136
statisticctransformnode properties	1136
statisticsmodelnode properties	1137
statisticsoutputnode Properties	1137
statisticsexportnode Properties	1138
Python Node Properties	1139
gmm properties	1139
hdbscannode properties	1139
kdemodel properties	1140
kdeexport properties	1118
gmm properties	1139
ocsvmnode properties	1142
rfnode properties	1142
smotendone Properties	1143
tsnenode Properties	1015
xgboostlinernode Properties	1144
xgboosttreinode Properties	1145
Spark Node Properties	1146
isotoniccasnode Properties	1146
kmeansasnnode properties	1045
multilayerperceptronnode Properties	1147
xgboostasnnode Properties	1147
SuperNode properties	1148
Scripting API Reference	
General Classes	1149
enum Objects	1149
ModelerException Objects	1150
VersionInfo Objects	1150
Content Management	1151
ColumnStatsContentModel Objects	1151
ContentModel Objects	1151
JSONContentModel Objects	1152
PairwiseStatsContentModel Objects	1152
StatisticType Objects	1153
TableContentModel Objects	1154
XMLContentModel Objects	1155
Core Objects	1156
ASCredentialDescriptor Objects	1157
ApplicationData Objects	1157
ContentFormat Objects	1158
ContentContainer Objects	1158
ContentContainerProvider Objects	1159
ContentProvider Objects	1160
CredentialDescriptor Objects	1161
FileFormat Objects	1161
IncompatibleServerException Objects	1163
InvalidPropertyException Objects	1164
ModelFieldRole Objects	1164

ObjectCreationException Objects	1165
ObjectLockedException Objects	1165
OwnerException Objects	1165
ParameterDefinition Objects	1166
ParameterProvider Objects	1167
ParameterStorage Objects	1168
ParameterType Objects	1168
PropertiedObject Objects	1169
RepositoryConnectionDescriptor Objects	1171
RepositoryConnectionDescriptor2 Objects	1171
RepositoryConnectionDescriptor3 Objects	1171
ServerConnectionDescriptor Objects	1172
ServerConnectionException Objects	1172
ServerVersionInfo Objects	1172
StructureAttributeType Objects	1173
StructuredValue Objects	1173
SystemServerConnectionDescriptor Objects	1174
Data and Metadata	1174
Column Objects	1175
ColumnCountException Objects	1176
ColumnGroup Objects	1177
ColumnGroupType Objects	1177
DataModel Objects	1178
DataModelError Objects	1180
DataModelFactory Objects	1181
ExtendedMeasure Objects	1185
ExtendedStorage Objects	1186
GeoType Objects	1186
GeometryType Objects	1186
GlobalValues Objects	1187
GlobalValues.Type Objects	1187
InvalidColumnExceptionValues Objects	1187
ListStorage Objects	1188
MeasureType Objects	1188
MissingValueDefinition Objects	1188
ModelOutputMetadata Objects	1189
ModelingRole Objects	1189
RowSet Objects	1190
StorageType Objects	1191
UnknownColumnException Objects	1191
Expressions	1191
Expression Objects	1192
Parser Objects	1192
ParserException Objects	1192
Model information	1193
CompositeModelDetail Objects	1193
ModelDetail Objects	1193
ModelType Objects	1194
PMMLModelType Objects	1195
Server resources	1196
ServerDatabaseConnection Objects	1196
ServerFile Objects	1197
ServerFileSystem Objects	1197
ServerResourceException Objects	1199
Server resources	1200
LocaleInfo Objects	1200
Repository Objects	1200
Session Objects	1203
SessionException Objects	1208
SystemSession Objects	1209
UIResources Objects	1209
Tasks and execution	1209

ExecutionFeedbackEvent Objects	1210
ExecutionFeedbackListener Objects	1211
ExecutionHandle Objects	1211
ExecutionState Objects	1212
ExecutionStateEvent Objects	1212
ExecutionStateListener Objects	1213
Task Objects	1213
TaskFactory Objects	1213
TaskRunner Objects	1216
Streams and SuperNodes	1221
BuiltObject Objects	1223
CFNode Objects	1223
CompositeModelApplier Objects	1223
CompositeModelBuilder Objects	1224
CompositeModelOutput Objects	1224
CompositeModelOwner Objects	1224
CompositeModelResults Objects	1225
SuperNode Objects	1226
SuperNodeDiagram Objects	1226
SuperNodeType Objects	1228
DataReader Objects	1228
DataTransformer Objects	1228
DataWriter Objects	1229
DiagramConnector Objects	1229
DocumentBuilder Objects	1229
DocumentOutput Objects	1229
DocumentOutputType Objects	1230
ExportFormatException Objects	1231
GraphBuilder Objects	1231
GraphOutput Objects	1231
InitialNode Objects	1231
ProcessNode Objects	1232
InvalidEditException Objects	1232
ModelApplier Objects	1232
ModelBuilder Objects	1232
ModelOutput Objects	1233
ModelOutputType Objects	1233
ObjectBuilder Objects	1235
Node Objects	1235
Diagram Objects	1236
NodeFilter Objects	1242
Stream Objects	1242
PublishedImage Objects	1244
ReportBuilder Objects	1244
ReportOutput Objects	1244
RowSetBuilder Objects	1244
RowSetOutput Objects	1245
TerminalNode Objects	1245
Updatable Objects	1245
Updater Objects	1246
In-Database Mining	
In-Database Mining	1246
Database Modeling Overview	1246
What you need	1247
Model Building	1247
Data Preparation	1247
Model scoring	1247
Exporting and saving database models	1248
Model Consistency	1248
Viewing and Exporting Generated SQL	1248
Database Modeling with Microsoft Analysis Services	1248
IBM SPSS Modeler and Microsoft Analysis Services	1248

Requirements for Integration with Microsoft Analysis Services	1249
Enabling Integration with Analysis Services	1250
Building Models with Analysis Services	1252
Managing Analysis Services Models	1252
Settings Common to All Algorithm Nodes	1253
Server Options	1254
Model Options	1254
MS Decision Tree Expert Options	1254
MS Clustering Expert Options	1255
MS Naive Bayes Expert Options	1255
MS Linear Regression Expert Options	1256
MS Neural Network Expert Options	1256
MS Logistic Regression Expert Options	1257
MS Association Rules Node	1257
MS Association Rules Expert Options	1258
MS Time Series Node	1258
MS Time Series Model Options	1259
MS Time Series Expert Options	1260
MS Time Series Settings Options	1260
MS Sequence Clustering Node	1261
MS Sequence Clustering Fields Options	1262
MS Sequence Clustering Expert Options	1262
Scoring Analysis Services Models	1263
Settings Common to All Analysis Services Models	1263
Analysis Services Model Nugget Server Tab	1264
Analysis Services Model Nugget Summary Tab	1264
MS Time Series Model Nugget	1264
MS Time Series Model Nugget Server Tab	1265
MS Time Series Model Nugget Settings Tab	1266
MS Sequence Clustering Model Nugget	1266
Exporting Models and Generating Nodes	1267
Analysis Services Mining Examples	1268
Example Streams: Decision Trees	1268
Example Stream: Upload Data	1269
Example Stream: Explore Data	1269
Example Stream: Build Model	1269
Example Stream: Evaluate Model	1269
Example Stream: Deploy Model	1270
Database Modeling with Oracle Data Mining	1271
About Oracle Data Mining	1271
Requirements for Integration with Oracle	1272
Enabling Integration with Oracle	1272
Building Models with Oracle Data Mining	1274
Oracle Models Server Options	1274
Misclassification Costs	1275
Oracle Naive Bayes	1276
Naive Bayes Model Options	1277
Naive Bayes Expert Options	1277
Oracle Adaptive Bayes	1278
Adaptive Bayes Model Options	1279
Adaptive Bayes Expert Options	1280
Oracle Support Vector Machine (SVM)	1281
Oracle SVM Model Options	1282
Oracle SVM Expert Options	1282
Oracle SVM Weights Options	1283
Oracle Generalized Linear Models (GLM)	1284
Oracle GLM Model Options	1285
Oracle GLM Expert Options	1285
Oracle GLM Weights Options	1286
Oracle Decision Tree	1286
Decision Tree Model Options	1287
Decision Tree Expert Options	1288

Oracle O-Cluster	1289
O-Cluster Model Options	1289
O-Cluster Expert Options	1290
Oracle k-Means	1291
K-Means Model Options	1291
K-Means Expert Options	1292
Oracle Nonnegative Matrix Factorization (NMF)	1293
NMF Model Options	1294
NMF Expert Options	1295
Oracle Apriori	1295
Apriori Fields Options	1296
Apriori Model Options	1297
Oracle Minimum Description Length (MDL)	1298
MDL Model Options	1298
Oracle Attribute Importance (AI)	1299
AI Model Options	1300
AI Selection Options	1301
AI Model Nugget Model Tab	1301
Managing Oracle Models	1302
Oracle Model Nugget Server Tab	1302
Oracle Model Nugget Summary Tab	1303
Oracle Model Nugget Settings Tab	1303
Listing Oracle Models	1304
Oracle Data Miner	1304
Preparing the Data	1305
Oracle Data Mining Examples	1305
Example Stream: Upload Data	1306
Example Stream: Explore Data	1307
Example Stream: Build Model	1307
Example Stream: Evaluate Model	1307
Example Stream: Deploy Model	1308
Database Modeling with IBM Data Warehouse and IBM Netezza Analytics	1308
SPSS Modeler with IBM Data Warehouse and IBM Netezza Analytics	1309
Integration requirements	1309
Enabling integration	1310
Configuring IBM Netezza Analytics or IBM Data Warehouse	1310
Creating an ODBC Source for IBM Netezza Analytics	1310
Enabling integration in SPSS Modeler	1311
Enabling SQL Generation and Optimization	1311
Building models with IBM Netezza Analytics and IBM Data Warehouse	1312
Field options	1312
Server options	1313
Model options	1313
Managing models	1313
Listing database models	1314
IBM Data WH Regression Tree	1314
IBM Data WH Regression Tree Build Options - Tree Growth	1314
IBM Data WH Tree Build Options - Tree Pruning	1314
Netezza Divisive Clustering	1315
Netezza Divisive Clustering Field Options	1315
Netezza Divisive Clustering Build Options	1316
IBM Data WH Generalized Linear	1316
IBM Data WH Generalized Linear Model Field Options	1317
IBM Data WH Generalized Linear Model Options - General	1317
IBM Data WH Generalized Linear Model Options - Interaction	1318
Add a Custom Term	1318
IBM Data WH Generalized Linear Model Options - Scoring Options	1318
IBM Data WH Decision Trees	1318
Instance weights and class weights	1319
Netezza Decision Tree Field Options	1319
IBM Data WH Decision Tree Build Options	1320
IBM Data WH Decision Tree Node - Class Weights	1320

IBM Data WH Decision Tree Node - Tree Pruning	1320
IBM Data WH Linear Regression	1321
IBM Data WH Linear Regression Build Options	1321
IBM Data WH KNN	1321
IBM Data WH KNN Model Options - General	1321
IBM Data WH KNN Model Options - Scoring Options	1322
IBM Data WH K-Means	1322
IBM Data WH K-Means Field Options	1323
IBM Data WH K-Means Build Options Tab	1323
IBM Data WH Naive Bayes	1323
Netezza Bayes Net	1324
Netezza Bayes Net Field Options	1324
Netezza Bayes Net Build Options	1324
Netezza Time Series	1324
Interpolation of Values in Netezza Time Series	1325
Netezza Time Series Field Options	1326
Netezza Time Series Build Options	1327
ARIMA Structure	1328
Netezza Time Series Build Options - Advanced	1328
Netezza Time Series Model Options	1328
IBM Data WH TwoStep	1329
IBM Data WH TwoStep Field Options	1329
IBM Data WH TwoStep Build Options	1329
IBM Data WH PCA	1330
IBM Data WH PCA Field Options	1330
IBM Data WH PCA Build Options	1330
Managing IBM Data WH and Netezza Models	1330
Scoring IBM Data Warehouse and IBM Netezza Analytics models	1331
IBM Data WH and Netezza model nugget Server tab	1331
IBM Data WH Decision Tree Model Nuggets	1331
IBM Data WH Decision Tree Nugget - Model Tab	1332
IBM Data WH Decision Tree Nugget - Settings Tab	1332
IBM Data WH Decision Tree Nugget - Viewer Tab	1332
IBM Data WH K-Means Model Nugget	1332
IBM Data WH K-Means Nugget - Model Tab	1333
IBM Data WH K-Means Nugget - Settings Tab	1333
Netezza Bayes Net Model Nuggets	1333
Netezza Bayes Net Nugget - Settings Tab	1333
IBM Data WH Naive Bayes Model Nuggets	1334
IBM Data WH Naive Bayes Nugget - Settings Tab	1334
IBM Data WH KNN Model Nuggets	1334
IBM Data WH KNN Nugget - Settings Tab	1335
Netezza Divisive Clustering Model Nuggets	1335
Netezza Divisive Clustering Nugget - Settings Tab	1335
IBM Data WH PCA Model Nuggets	1336
IBM Data WH PCA Nugget - Settings Tab	1336
Netezza Regression Tree Model Nuggets	1336
Netezza Regression Tree Nugget - Model Tab	1337
Netezza Regression Tree Nugget - Settings Tab	1337
Netezza Regression Tree Nugget - Viewer Tab	1337
IBM Data WH Linear Regression Model Nuggets	1338
IBM Data WH Linear Regression Nugget - Settings Tab	1338
Netezza Time Series Model Nugget	1338
Netezza Time Series Nugget - Settings tab	1338
IBM Data WH Generalized Linear Model Nugget	1338
IBM Data WH Generalized Linear Model Nugget - Settings tab	1339
IBM Data WH TwoStep Model Nugget	1339
IBM Data WH TwoStep Nugget - Model Tab	1339
Database modeling with IBM Db2 for z/OS	1339
IBM SPSS Modeler and IBM Db2 for z/OS	1340
Requirements for integration with IBM Db2 for z/OS	1340
Enabling integration with IBM Db2 Analytics Accelerator for z/OS	1340

Configuring IBM Db2 for z/OS and IBM Analytics Accelerator for z/OS	1340
Creating an ODBC Source for IBM Db2 for z/OS and IBM Db2 Analytics Accelerator	1341
Enabling the integration of IBM Db2 for z/OS in IBM SPSS Modeler	1341
Enabling SQL Generation and Optimization	1341
Configuring DSN using IBM Db2 Client in IBM SPSS Modeler	1342
Building models with IBM Db2 for z/OS	1342
IBM Db2 for z/OS models - Field options	1343
IBM Db2 for z/OS Models - Server Options	1343
IBM Db2 for z/OS models - Model options	1343
IBM Db2 for z/OS Models - K-Means	1344
IBM Db2 for z/OS models - K-Means Field options	1344
IBM Db2 for z/OS Models - K-Means build options	1344
IBM Db2 for z/OS models - Naive Bayes	1345
IBM Db2 for z/OS Models - Decision Trees	1345
IBM Db2 for z/OS models - Decision Tree field options	1345
IBM Db2 for z/OS Models - Decision Tree Build Options	1345
IBM Db2 for z/OS Models - Decision Tree Node - Class Weights	1346
IBM Db2 for z/OS Models - Decision Tree Node - Tree Pruning	1346
IBM Db2 for z/OS models - Regression Tree	1346
IBM Db2 for z/OS Models - Regression Tree Build Options - Tree Growth	1346
IBM Db2 for z/OS models - Regression Tree build options - Tree Pruning	1347
IBM Db2 for z/OS models - TwoStep	1347
IBM Db2 for z/OS models - TwoStep field options	1348
IBM Db2 for z/OS Models - TwoStep Build Options	1348
IBM Db2 for z/OS Models - TwoStep nugget - Model tab	1348
Managing IBM Db2 for z/OS Models	1348
Scoring IBM Db2 for z/OS Models	1349
IBM Db2 for z/OS Decision Tree Model Nuggets	1349
IBM Db2 for z/OS Decision Tree Nugget - Model Tab	1349
IBM Db2 for z/OS Decision Tree Nugget - Viewer Tab	1350
IBM Db2 for z/OS K-Means model nugget	1350
IBM Db2 for z/OS K-Means nugget - Model tab	1350
IBM Db2 for z/OS Naive Bayes model nuggets	1350
IBM Db2 for z/OS Regression Tree model nuggets	1350
IBM Db2 for z/OS Regression Tree nugget - Model tab	1350
IBM Db2 for z/OS Regression Tree nugget - Viewer tab	1351
IBM Db2 for z/OS TwoStep model nugget	1351
Server Administration and Performance	
Architecture and Hardware Recommendations	1351
Architecture Description	1351
Hardware Recommendations	1352
Temporary Disk Space and RAM Requirements	1353
Conditions That Require Temporary Disk Space	1353
Calculating the Amount of Temporary Disk Space	1353
RAM Requirements	1354
Data Access	1354
Referencing Data Files	1355
Importing IBM SPSS Statistics Data Files	1355
IBM SPSS Modeler Support	1355
Connecting to IBM SPSS Modeler Server	74
Configuring single sign-on	1357
The Service Principal Name	1357
Configuring IBM SPSS Modeler Server on Windows	1358
Configuring IBM SPSS Modeler Server on UNIX and Linux	1359
Configuring IBM SPSS Modeler client	1359
Getting the SSO user's group membership	1360
Single sign-on for data sources	1361
Adding and Editing the IBM SPSS Modeler Server Connection	75
Searching for Servers in IBM SPSS Collaboration and Deployment Services	75
Data and File Systems	1362
User Authentication	1362
Permissions	1363

File Creation	1363
Differences in Results	1363
IBM SPSS Modeler Administration	1364
Starting and Stopping IBM SPSS Modeler Server	1364
To Start, Stop, and Check Status on Windows	1364
To Start, Stop, and Check Status on UNIX	1364
Handling Unresponsive Server Processes (UNIX Systems)	1365
Configuring server profiles	1365
Working with server profiles	1366
Profile structure	1367
Profile scripts	1369
Common script (for all platforms)	1369
Windows scripts	1370
UNIX script	1371
Administration	1372
IBM SPSS Modeler Server Administration	1372
Starting Modeler Administration Console	1373
Restarting the web service	1373
Configuring Access with Modeler Administration Console	1373
Configuring Access with User Access Control	1374
SPSS Modeler Server connections	1374
Server Login Dialog Box	1375
SPSS Modeler Server Configuration	1375
Connections/Sessions	1375
Analytic Server connection	1375
Data file access	1376
Performance/Optimization	1377
SQL	1378
SSL	1378
Coordinator of Processes configuration	1378
Options visible in options.cfg	1379
SPSS Modeler Server Monitoring	1380
Using the options.cfg file	1381
Closing unused database connections	1381
Using SSL to secure data transfer	1382
How SSL works	1382
Securing client/server and server-server communications with SSL	1382
Obtaining and installing SSL certificate and keys	1383
Configuring the environment to run GSKit	1383
Creating an SSL key database	1384
Creating a self-signed SSL certificate	1384
Installing a self-signed SSL certificate	1384
Importing a third-party root CA certificate	1385
Enable and configure SSL in IBM SPSS Deployment Manager	1385
Installing unlimited strength encryption	1385
Instructing users to enable SSL	1386
Cognos SSL connection	1386
Cognos TM1 SSL connection	1386
Configuring the URL prefix	1387
Securing LDAP with SSL	1387
Configuring groups	1388
Log files	1390
Performance Overview	1391
Server performance and optimization settings	1391
Client Performance and Optimization Settings	1391
Database Usage and Optimization	1392
SQL Optimization	1392
Stream performance	1393
SQL optimization	1393
How SQL generation works	1394
SQL Generation Example	1395
Configuring SQL Optimization	1396

Previewing Generated SQL	1396
Viewing SQL for Model Nuggets	1397
Tips for maximizing SQL generation	1397
Nodes supporting SQL generation	1398
CLEM Expressions and Operators Supporting SQL Generation	1400
Using SQL Functions in CLEM Expressions	1402
Writing SQL Queries	1402
Scoring adapter for Teradata - duplicate rows	1402
Configuring Oracle for UNIX Platforms	1403
Configuring Oracle for SQL Optimization	1403
Configuring UNIX Startup Scripts	1404
Introduction	1404
Scripts	1404
Automatically Starting and Stopping IBM SPSS Modeler Server	1404
Manually Starting and Stopping IBM SPSS Modeler Server	1405
Editing Scripts	1405
Controlling Permissions on File Creation	1405
IBM SPSS Modeler Server and the data access pack	1405
Troubleshooting ODBC configuration	1407
Library paths	1409
Configuring and Running SPSS Modeler Server as a Non-Root Process on UNIX	1409
Introduction	1409
Configuring as non-root without a private password database	1410
Configuring as non-root using a private password database	1410
Running SPSS Modeler Server as a non-root user	1411
Troubleshooting user authentication failures	1411
Configuring and Running SPSS Modeler Server with a private password file on Windows	1411
Introduction	1412
Configuring a private password database	1412
Load Balancing with Server Clusters	1412
LDAP authentication	1413
Deployment Guide	
Overview	1414
IBM SPSS Collaboration and Deployment Services	1414
Collaboration	1415
Deployment	1415
System architecture	1415
IBM SPSS Collaboration and Deployment Services Repository	1417
IBM SPSS Modeler with collaboration	1418
IBM SPSS Deployment Manager	1418
IBM SPSS Collaboration and Deployment Services Deployment Portal	1418
Browser-based IBM SPSS Deployment Manager	1419
Execution servers	1419
Working with files	1420
Server definitions	1420
Adding new server definitions	1420
IBM SPSS Modeler server parameters	1421
Modifying server definitions	1421
IBM SPSS Modeler Job Steps	1421
Working with IBM SPSS Modeler streams	1421
IBM SPSS Modeler Server configuration	1422
Viewing IBM SPSS Modeler job properties	1422
IBM SPSS Modeler Job Properties - General	1422
IBM SPSS Modeler Job Properties - Data Files	1424
Specifying the input file location	1425
IBM SPSS Modeler Job Properties - Data View	1425
IBM SPSS Modeler Job Properties - ODBC data sources	1425
Changing the ODBC connection	1426
Browsing for an ODBC connection	1426
Changing the database credentials	1426
Browsing for a credential definition	1426
IBM SPSS Modeler Job Properties - Geo Spatial	1426
IBM SPSS Modeler Job Properties - Parameters	1427

IBM SPSS Modeler Job Properties - Results	1427
Viewing output results	1428
IBM SPSS Modeler Job properties - Cognos Import	1428
IBM SPSS Modeler Job properties - Cognos Export	1428
IBM SPSS Modeler Job Properties - Legacy TM1 Import	1429
IBM SPSS Modeler Job Properties - Legacy TM1 Export	1429
IBM SPSS Modeler Job Properties - TM1 Import	1429
IBM SPSS Modeler Job Properties - TM1 Export	1429
IBM SPSS Modeler Job Properties - Analytic Server Import	1430
IBM SPSS Modeler Job Properties - Analytic Server Export	1430
IBM SPSS Modeler Job Properties - Notifications	1430
Viewing Streams in IBM SPSS Modeler	1430
IBM SPSS Modeler Completion Codes	1431
IBM SPSS Modeler Stream Limitations	1431
Node Types	1432
Scoring Service	1432
IBM SPSS Modeler stream limitations	1432
IBM SPSS Modeler Extensions Help	
Supported languages	1432
R	1433
Python for Spark	1433
Scripting with Python for Spark	1433
Analytic Server Context	1436
Data metadata	1438
Date, time, timestamp	1439
Exceptions	1439
Examples	1440
Switching to a different Python environment	1442
Extension nodes	1442
Extension Export node	1442
Extension Export node - Syntax tab	1443
Extension Export node - Console Output tab	1443
Publishing streams	1443
Extension Output node	1444
Extension Output node - Syntax tab	1444
Extension Output node - Console Output tab	1445
Extension Output node - Output tab	1445
Extension Output Browser	1445
Extension Output Browser - Text Output tab	1446
Extension Output Browser - Graph Output tab	1446
Extension Model node	1446
Extension Model node - Syntax tab	1446
Extension Model node - Model Options tab	1446
Extension Model node - Console Output tab	1446
Extension Model node - Text Output tab	1447
Extension model nugget	1447
Extension model nugget - Syntax tab	1447
Extension model nugget - Model Options tab	1447
Extension model nugget - Graph Output tab	1448
Extension model nugget - Text Output tab	1448
Extension model nugget - Console Output tab	1448
Extension Transform node	1449
Extension Transform node - Syntax tab	1449
Extension Transform node - Console Output tab	1449
Extension Import node	1449
Extension Import node - Syntax tab	1450
Extension Import node - Console Output tab	1450
Filtering or renaming fields	1450
Viewing and setting information about types	1451
Extensions	1451
Extension Hub	1451
Explore tab (Extension Hub)	1452

How to get integration plug-ins	1452
Installed tab (Extension Hub)	1452
Settings (Extension Hub)	1453
Extension Details	1453
Installing local extension bundles	1454
Installation locations for extensions	1454
Required R packages	1455
Creating and managing custom nodes	1455
Custom Dialog Builder layout	1456
Building a custom node dialog	1456
Dialog Properties	1457
Laying out controls on the dialog canvas	1457
Building the script template	1458
Previewing a custom node dialog	1459
Control types	1459
Field Chooser	1460
Specifying the Field Source for a Field Chooser	1461
Filtering Field Lists	1461
Check Box	1462
Combo Box	1462
Specifying list items for combo boxes and list boxes	1463
List Box	1464
Text control	1465
Number Control	1466
Date control	1467
Secured Text	1468
Static Text Control	1469
Color Picker	1470
Table Control	1471
Specifying columns for table controls	1472
Item Group	1473
Radio Group	1474
Defining radio buttons	1474
Check Box Group	1475
File Browser	1476
File type filter	1477
Tab	1477
Sub-dialog button	1478
Dialog properties for a sub-dialog	1479
Specifying an Enabling Rule for a Control	1479
Extension Properties	1480
Required properties of extensions	1480
Optional properties of extensions	1481
Managing custom node dialogs	1482
Creating Localized Versions of Custom Node Dialogs	1484
Importing and exporting data using Python for Spark	1485
Importing and exporting data using R	1485

IBM SPSS Modeler CRISP-DM Guide

Introduction to CRISP-DM	1485
CRISP-DM Help Overview	1486
CRISP-DM in IBM SPSS Modeler	1486
CRISP-DM Project Tool	1487
Help for CRISP-DM	1487
Additional resources	1487
Business Understanding	1488
Business Understanding Overview	1488
Determining Business Objectives	1488
E-Retail Example--Finding Business Objectives	1488
Compiling the Business Background	1489
Defining Business Objectives	1489
Business Success Criteria	1490
Assessing the Situation	1490

E-Retail Example--Assessing the Situation	1490
Resource Inventory	1491
Requirements, Assumptions, and Constraints	1491
Risks and Contingencies	1492
Terminology	1492
Cost/Benefit Analysis	1493
Determining Data Mining Goals	1493
Data Mining Goals	1493
E-Retail Example--Data Mining Goals	1494
Data Mining Success Criteria	1494
Producing a Project Plan	1494
Writing the Project Plan	1494
Sample Project Plan	1495
Assessing Tools and Techniques	1495
Ready for the next step?	1495
Data Understanding	1496
Data Understanding Overview	1496
Collecting Initial Data	1496
E-Retail Example--Initial Data Collection	1497
Writing a Data Collection Report	1497
Describing Data	1497
E-Retail Example--Describing Data	1498
Writing a Data Description Report	1498
Exploring Data	1498
E-Retail Example--Exploring Data	1499
Writing a Data Exploration Report	1499
Verifying Data Quality	1499
E-Retail Example--Verifying Data Quality	1500
Writing a Data Quality Report	1500
Ready for the next step?	1500
Data Preparation	1501
Data Preparation Overview	1501
Selecting Data	1501
E-Retail Example--Selecting Data	1502
Including or Excluding Data	1502
Cleaning Data	1502
E-Retail Example--Cleaning Data	1502
Writing a Data Cleaning Report	1503
Constructing New Data	1503
E-Retail Example--Constructing Data	1503
Deriving Attributes	1504
Integrating Data	1504
E-Retail Example--Integrating Data	1504
Integration Tasks	1505
Formatting Data	1505
Ready for modeling?	1505
Modeling	1505
Modeling Overview	1506
Selecting Modeling Techniques	1506
E-Retail Example--Modeling Techniques	1506
Choosing the Right Modeling Techniques	1507
Modeling Assumptions	1507
Generating a Test Design	1507
Writing a Test Design	1508
E-Retail Example--Test Design	1508
Building the Models	1508
E-Retail Example--Model Building	1509
Parameter Settings	1509
Running the Models	1509
Model description	1509
Assessing the Model	1510
Comprehensive Model Assessment	1510
E-Retail Example--Model Assessment	1510
Keeping Track of Revised Parameters	1511

Ready for the next step?	1511
Evaluation	1511
Evaluation Overview	1511
Evaluating the Results	1512
E-Retail Example--Evaluating Results	1512
Review Process	1512
E-Retail Example--Review Report	1513
Determining the Next Steps	1513
E-Retail Example--Next Steps	1513
Deployment	1513
Deployment Overview	1514
Planning for Deployment	1514
E-Retail Example--Deployment Planning	1515
Planning Monitoring and Maintenance	1515
E-Retail Example--Monitoring and Maintenance	1515
Producing a Final Report	1516
Preparing a Final Presentation	1516
E-Retail Example--Final Report	1516
Conducting a Final Project Review	1516
E-Retail Example--Final Review	1517

IBM SPSS Modeler Messages

AEQMC0000E	1517
AEQMC0001E	1517
AEQMC0002E	1517
AEQMC0003E	1517
AEQMC0004E	1517
AEQMC0005E	1517
AEQMC0006E	1518
AEQMC0007I	1518
AEQMC0008I	1518
AEQMC0009I	1518
AEQMC0010I	1518
AEQMC0011I	1518
AEQMC0012I	1518
AEQMC0013I	1518
AEQMC0014I	1518
AEQMC0015E	1518
AEQMC0016I	1518
AEQMC0017E	1518
AEQMC0018I	1518
AEQMC0019I	1518
AEQMC0020I	1518
AEQMC0021E	1518
AEQMC0022E	1519
AEQMC0023I	1519
AEQMC0024I	1519
AEQMC0025I	1519
AEQMC0026I	1519
AEQMC0027I	1519
AEQMC0028I	1519
AEQMC0028I	1519
AEQMC0029I	1519
AEQMC0030I	1519
AEQMC0031E	1519
AEQMC0031E	1519
AEQMC0032I	1519
AEQMC0033E	1519
AEQMC0034I	1519
AEQMC0035I	1519
AEQMC0035E	1519
AEQMC0036E	1519
AEQMC0037E	1520
AEQMC0038I	1520
AEQMC0039I	1520
AEQMC0040I	1520
AEQMC0041I	1520
AEQMC0042I	1520

AEQMC0043I	1520
AEQMC0044E	1520
AEQMC0045E	1520
AEQMC0046E	1520
AEQMC0047E	1520
AEQMC0048E	1520
AEQMC0049I	1520
AEQMC0050I	1520
AEQMC0051I	1520
AEQMC0052I	1521
AEQMC0053I	1521
AEQMC0054I	1521
AEQMC0055I	1521
AEQMC0056I	1521
AEQMC0057I	1521
AEQMC0058I	1521
AEQMC0059I	1521
AEQMC0060I	1521
AEQMC0061I	1521
AEQMC0062E	1521
AEQMC0063I	1521
AEQMC0064E	1521
AEQMC0065E	1521
AEQMC0066E	1521
AEQMC0067E	1522
AEQMC0068E	1522
AEQMC0069E	1522
AEQMC0070I	1522
AEQMC0071I	1522
AEQMC0072I	1522
AEQMC0073E	1522
AEQMC0074E	1522
AEQMC0075E	1522
AEQMC0076I	1522
AEQMC0077E	1522
AEQMC0078E	1522
AEQMC0079I	1522
AEQMC0080E	1522
AEQMC0081I	1522
AEQMC0082E	1523
AEQMC0083E	1523
AEQMC0084E	1523
AEQMC0085E	1523
AEQMC0086E	1523
AEQMC0087E	1523
AEQMC0088E	1523
AEQMC0089E	1523
AEQMC0090E	1523
AEQMC0091I	1523
AEQMC0092E	1523
AEQMC0093E	1523
AEQMC0094E	1523
AEQMC0095I	1523
AEQMC0096I	1523
AEQMC0097I	1523
AEQMC0098I	1524
AEQMC0099E	1524
AEQMC0132I	1524
AEQMJ0002I	1524
AEQMJ0003E	1524
AEQMJ0004E	1524
AEQMJ0005E	1524

IBM SPSS Modeler Text Analytics Help

About IBM SPSS Modeler Text Analytics	1524
Upgrading IBM SPSS Modeler Text Analytics	1525
About text mining	1525
How extraction works	1527

How categorization works	1528
IBM SPSS Modeler Text Analytics nodes	1529
Applications	1529
Reading in Source Text	1530
File List node	1530
File List node: Settings tab	1530
File List Node: Other Tabs	1531
Using the File List node in text mining	1531
Web Feed node	1532
Web Feed Node: Input Tab	1532
Web Feed Node: Records Tab	1533
Web Feed Node: Content Filter Tab	1533
Using the Web Feed Node in Text Mining	1534
Language Node	1535
Language Node: Settings Tab	1535
Mining for Concepts and Categories	1535
Text Mining modeling node	1536
Text Mining Node: Fields Tab	1537
Document Settings for Fields Tab	1538
Text Mining Node: Model Tab	1539
Build Interactively	1540
Generate Directly	1541
Copying resources from templates and TAPs	1541
Text Mining node: Expert tab	1542
Sampling Upstream to Save Time	1543
Using the Text Mining node in a stream	1544
Text Mining Nugget: Concept Model	1545
Concept Model: Model Tab	1545
Options for Including Concepts for Scoring	1546
Underlying Terms in Concept Models	1547
Concept model: Settings tab	1547
Concept Model: Fields tab	1548
Concept Model: Summary Tab	1549
Using Concept Model Nuggets in a Stream	1549
Text Mining Nugget: Category Model	1551
Category Model Nugget: Model Tab	1552
Category model nugget: Settings tab	1553
Category Model Nugget: Other Tabs	1554
Using category model nuggets in a stream	1554
Mining for Text Links	1556
Text Link Analysis node	1556
Text Link Analysis node: Fields tab	1557
Text Link Analysis node: Expert tab	1558
TLA node output	1559
Caching TLA Results	1559
Using the Text Link Analysis node in a stream	1559
Browsing External Source Text	1561
File Viewer node	1561
File Viewer Node Settings	1561
Using the File Viewer Node	1561
Node Properties for Scripting	1563
File List Node: filelistnode	1564
Web Feed Node: webfeednode	1564
Language Node: languageidentifier	1565
Text Mining node: TextMiningWorkbench	1565
Text Mining model nugget: TMWBMModelApplier	1566
Text Link Analysis node: textlinkanalysis	1567
Interactive workbench mode	1568
The Categories and Concepts View	1569
The Clusters View	1570
The Text Link Analysis view	1571
The Resource Editor view	1573
Setting Options	1573
Options: Session Tab	1574

Options: Display Tab	1574
Options: Sounds Tab	1575
Microsoft Internet Explorer settings for Help	1575
Generating Model Nuggets and Modeling Nodes	1576
Updating Modeling Nodes and Saving	1576
Closing and Ending Sessions	1576
Keyboard Accessibility	1577
Shortcuts for Dialog Boxes	1578
Extracting Concepts and Types	1578
Extraction results: Concepts and types	1579
Extracting data	1579
Filtering Extraction Results	1581
Exploring concept maps	1582
Building Concept Map Indexes	1583
Refining extraction results	1584
Adding synonyms	1585
Adding concepts to types	1585
Excluding concepts from extraction	1586
Forcing Words into Extraction	1586
Categorizing text data	1587
The Categories Pane	1588
Methods and Strategies for Creating Categories	1589
Methods for Creating Categories	1590
Strategies for Creating Categories	1590
Tips for Creating Categories	1591
Choosing the Best Descriptors	1592
About Categories	1594
Category Properties	1594
The Data Pane	1595
Category Relevance	1596
Flagging responses	1596
Building categories	1597
Advanced linguistic settings	1598
Managing Link Exception Pairs	1599
About linguistic techniques	1600
Concept root derivation	1600
Concept Inclusion	1601
Semantic Networks	1602
Co-occurrence Rules	1603
Advanced Frequency Settings	1603
Extending categories	1604
Creating Categories Manually	1606
Creating New or Renaming Categories	1607
Creating Categories by Drag-and-Drop	1607
Using Category Rules	1607
Category Rule Syntax	1608
Using TLA Patterns in Category Rules	1609
Using Wildcards in Category Rules	1611
Category Rule Examples	1612
Creating Category Rules	1613
Editing and Deleting Rules	1614
Importing and Exporting Predefined Categories	1614
Importing Predefined Categories	1615
Flat List Format	1616
Compact Format	1616
Indented Format	1617
Exporting Categories	1618
Using Text Analysis Packages	1619
Making Text Analysis Packages	1619
Loading Text Analysis Packages	1620
Updating Text Analysis Packages	1620
Editing and Refining Categories	1621
Adding Descriptors to Categories	1621
Editing Category Descriptors	1622

Moving Categories	1622
Flattening Categories	1623
Merging or Combining Categories	1623
Forcing documents into categories	1623
Deleting Categories	1624
Analyzing clusters	1624
Building Clusters	1625
Calculating Similarity Link Values	1626
Exploring Clusters	1627
Cluster Definitions	1628
Exploring Text Link Analysis	1628
Extracting TLA Pattern Results	1629
Type and Concept Patterns	1630
Filtering TLA Results	1631
Data Pane	1632
Flagging responses	1596
Type Reassignment Rules	1633
Visualizing Graphs	1635
Category Graphs and Charts	1635
Category Bar Chart	1636
Category Web Graph	1637
Category Web Table	1637
Fixing Errors	1637
Cluster Graphs	1637
Concept Web Graph	1638
Cluster Web Graph	1638
Text Link Analysis Graphs	1639
Concept Web Graph	1639
Type Web Graph	1640
Using Graph Toolbars and Palettes	1640
Session Resource Editor	1641
Editing resources in the Resource Editor	1642
Making and Updating Templates	1642
Switching resource templates	1643
Templates and Resources	1643
Template Editor vs. Resource Editor	1644
The Editor interface	1645
Opening Templates	1647
Saving Templates	1647
Updating Node Resources After Loading	1648
Managing Templates	1649
Importing and Exporting Templates	1650
Exiting the Template Editor	1650
Backing Up Resources	1650
Importing resource files	1651
Working with Libraries	1652
Shipped libraries	1653
Creating Libraries	1653
Adding public libraries	1654
Finding Terms and Types	1654
Viewing Libraries	1655
Managing Local Libraries	1655
Renaming Local Libraries	1656
Disabling Local Libraries	1656
Deleting Local Libraries	1656
Managing Public Libraries	1657
Sharing Libraries	1657
Publishing Libraries	1658
Updating Libraries	1659
Resolving Conflicts	1659
About Library Dictionaries	1660
Type dictionaries	1660
Built-in types	1661
Creating types	1662
Adding terms	1662
Forcing terms	1664

Renaming types	1664
Moving types	1664
Disabling and deleting types	1665
Substitution/Synonym dictionaries	1665
Defining synonyms	1666
Defining optional elements	1667
Disabling and Deleting Substitutions	1667
Exclude dictionaries	1667
About Advanced Resources	1668
Finding	1669
Replacing	1670
Target Language for Resources	1670
Fuzzy Grouping	1671
Nonlinguistic Entities	1671
Regular Expression Definitions	1672
Normalization	1674
Configuration	1674
Language Handling	1675
Extraction patterns	1676
Forced Definitions	1677
Abbreviations	1678
About Text Link Rules	1678
Where to work on text link rules	1679
Where to Begin	1679
When to Edit or Create Rules	1680
Simulating Text Link Analysis Results	1680
Defining Data for Simulation	1681
Understanding Simulation Results	1681
Navigating Rules and Macros in the Tree	1682
Working with Macros	1683
Creating and Editing Macros	1684
Disabling and Deleting Macros	1685
Checking for Errors, Saving, and Cancelling	1685
Special Macros: mTopic, mNonLingEntities, SEP	1686
Working with Text Link Rules	1687
Creating and Editing Rules	1689
Disabling and Deleting Rules	1689
Checking for Errors, Saving, and Cancelling	1690
Processing Order for Rules	1690
Working with Rule Sets (Multiple Pass)	1691
Supported Elements for Rules and Macros	1692
Viewing and working in source mode	1694

IBM SPSS Modeler Tutorial

About IBM SPSS Modeler	1696
IBM SPSS Modeler Products	1696
IBM SPSS Modeler	1697
IBM SPSS Modeler Server	1697
IBM SPSS Modeler Administration Console	1697
IBM SPSS Modeler Batch	1697
IBM SPSS Modeler Solution Publisher	1697
IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services	1698
IBM SPSS Modeler Editions	1698
Documentation	1698
SPSS Modeler Professional documentation	1698
SPSS Modeler Premium Documentation	1699
Application examples	1699
Demos Folder	1700
License tracking	1700
Product overview	1700
Getting started	1700
Starting IBM SPSS Modeler	1700
IBM SPSS Modeler Interface at a Glance	1701
IBM SPSS Modeler Stream Canvas	1702
Nodes palette	1702

IBM SPSS Modeler Managers	1703
IBM SPSS Modeler Projects	1704
Using the Mouse in IBM SPSS Modeler	1705
Introduction to Modeling	1706
Building the Stream	1707
Browsing the Model	1710
Evaluating the Model	1712
Scoring records	1714
Summary	1715
Automated Modeling for a Flag Target	1715
Modeling Customer Response (Auto Classifier)	1715
Historical Data	1715
Building the Stream	1716
Generating and Comparing Models	1718
Summary	1722
Automated Modeling for a Continuous Target	1722
Property Values (Auto Numeric)	1722
Training Data	1723
Building the Stream	1723
Comparing the Models	1725
Summary	1727
Automated Data Preparation (ADP)	1727
Building the stream	1728
Comparing Model Accuracy	1730
Preparing Data for Analysis (Data Audit)	1732
Building the Stream	1732
Browsing Statistics and Charts	1734
Handling Outliers and Missing Values	1736
Drug Treatments (Exploratory Graphs/C5.0)	1738
Reading in Text Data	1739
Adding a Table	1741
Creating a Distribution Graph	1742
Creating a Scatterplot	1744
Creating a Web Graph	1745
Deriving a New Field	1746
Building a Model	1748
Browsing the model	1749
Using an Analysis Node	1750
Screening Predictors (Feature Selection)	1751
Building the Stream	1751
Building the Models	1754
Comparing the Results	1755
Summary	1755
Reducing Input Data String Length (Reclassify Node)	1756
Reducing Input Data String Length (Reclassify)	1756
Reclassifying the Data	1756
Modeling Customer Response (Decision List)	1759
Historical Data	1759
Building the Stream	1760
Creating the model	1761
Calculating custom measures using Excel	1769
Modifying the Excel template	1772
Saving the Results	1773
Classifying Telecommunications Customers (Multinomial Logistic Regression)	1774
Building the Stream	1774
Browsing the Model	1776
Telecommunications Churn (Binomial Logistic Regression)	1778
Building the Stream	1779
Browsing the Model	1783
Forecasting Bandwidth Utilization (Time Series)	1786
Forecasting with the Time Series Node	1786
Creating the Stream	1787
Examining the Data	1787
Defining the Dates	1789
Defining the Targets	1790
Setting the Time Intervals	1791
Creating the Model	1792

Examining the model	1793
Summary	1796
Reapplying a Time Series Model	1796
Retrieving the Stream	1797
Retrieving the saved model	1798
Generating a Modeling Node	1798
Generating a New Model	1798
Examining the New Model	1799
Summary	1801
Forecasting Catalog Sales (Time Series)	1801
Creating the Stream	1802
Examining the Data	1803
Exponential Smoothing	1803
ARIMA	1807
Summary	1809
Making Offers to Customers (Self-Learning)	1809
Building the Stream	1810
Browsing the model	1812
Predicting Loan Defaulters (Bayesian Network)	1815
Building the Stream	1816
Browsing the model	1818
Retraining a Model on a Monthly Basis (Bayesian Network)	1820
Building the Stream	1821
Evaluating the model	1823
Retail Sales Promotion (Neural Net/C&RT)	1826
Examining the Data	1827
Learning and Testing	1828
Condition Monitoring (Neural Net/C5.0)	1829
Examining the Data	1830
Data Preparation	1831
Learning	1832
Testing	1832
Classifying Telecommunications Customers (Discriminant Analysis)	1833
Creating the Stream	1833
Examining the Model	1836
Analyzing output of using Discriminant analysis to classify telecommunications customers	1836
Stepwise Discriminant analysis	1836
A note of caution concerning stepwise methods	1838
Checking model fit	1838
Structure matrix	1839
Territorial map	1840
Classification results	1841
Summary	1842
Analyzing interval-censored survival data (Generalized Linear Models)	1842
Creating the Stream	1843
Tests of model effects	1845
Fitting the treatment-only model	1845
Parameter estimates	1846
Predicted recurrence and survival probabilities	1846
Modeling the recurrence probability by period	1849
Tests of model effects	1852
Fitting the reduced model	1852
Parameter estimates	1853
Predicted recurrence and survival probabilities	1853
Summary	1855
Related procedures	1856
Recommended readings	1856
Using Poisson regression to analyze ship damage rates (Generalized Linear Models)	1856
Fitting an "overdispersed" Poisson regression	1856
Goodness-of-fit statistics	1859
Omnibus test	1860
Tests of model effects	1860
Parameter estimates	1861
Fitting alternative models	1861
Goodness-of-fit statistics	1862
Summary	1863
Related procedures	1856

Recommended readings	1856
Fitting a Gamma regression to car insurance claims (Generalized Linear Models)	1863
Creating the Stream	1864
Parameter estimates	1866
Summary	1866
Related procedures	1856
Recommended readings	1856
Classifying Cell Samples (SVM)	1867
Creating the Stream	1867
Examining the Data	1870
Trying a Different Function	1872
Comparing the Results	1872
Summary	1873
Using Cox Regression to Model Customer Time to Churn	1874
Building a Suitable Model	1874
Censored Cases	1876
Categorical Variable Codings	1877
Variable Selection	1878
Covariate Means	1879
Survival Curve	1880
Hazard Curve	1880
Evaluation	1881
Tracking the Expected Number of Customers Retained	1884
Scoring	1890
Summary	1894
Market Basket Analysis (Rule Induction/C5.0)	1894
Accessing the Data	1894
Discovering affinities in basket contents	1895
Profiling the Customer Groups	1897
Summary	1898
Assessing New Vehicle Offerings (KNN)	1898
Creating the Stream	1899
Examining the output	1901
Predictor Space	1902
Peers Chart	1902
Neighbor and Distance Table	1904
Summary	1904
Uncovering causal relationships in business metrics (TCM)	1904
Creating the stream	1904
Running the analysis	1905
Overall Model Quality Chart	1906
Overall Model System	1907
Impact Diagrams	1909
Determining root causes of outliers	1910
Running scenarios	1912
Glossary	1916
IBM SPSS Modeler Gold	1916
Overview	1916
Prerequisites	1917
Installing IBM SPSS Modeler Gold	1918
IBM SPSS Collaboration and Deployment Services Server components (Server 1)	1918
IBM SPSS Modeler Server components (Server 2)	1919
Client components	1919
Optional components	1919
Post-installation steps	1919
IBM SPSS Collaboration and Deployment Services Server components (Server 1)	1920
IBM SPSS Modeler Server components (Server 2)	1920
Client components	1920
Other	1920
Passport Advantage part numbers	1920
GDPR	1921

IBM SPSS Modeler V18.4.0 documentation

Welcome to the IBM® SPSS® Modeler documentation, where you can find information about how to install and use SPSS Modeler.

While IBM values the use of inclusive language, terms that are outside of IBM's direct influence are sometimes required for the sake of maintaining user understanding. As other industry leaders join IBM in embracing the use of inclusive language, IBM will continue to update the documentation to reflect those changes.

Getting started

- [Product overview](#)
- [What's new?](#)
- [Getting started](#)
- [Product requirements](#)

Common tasks

- [Building data streams](#)
- [Using a repository](#)
- [Creating a project](#)
- [Mining for concepts and categories using Text Analytics](#)

Troubleshooting and support

- [IBM Support](#)
- [IBM SPSS Support](#)
- [Fixes and downloads](#)
- [DeveloperWorks](#)

More information

- [Tutorials](#)
- [Algorithms Guide](#)
- [Manuals in PDF format](#)
- [Articles](#)
- [Redbooks](#)

© Copyright IBM Corporation 1994, 2022

Installation instructions

The following instructions are for installing IBM® SPSS® Modeler version 18.4.0 using the license type concurrent license. This document is for users who are installing on their desktop computers.

- [System requirements](#)
- [Installing](#)
- [Licensing your product](#)
- [Configuring IBM SPSS Modeler to work with IBM SPSS Statistics](#)
- [Database Access](#)
- [Checking out/in a commuter license](#)
- [Applying fix packs](#)
- [Uninstalling](#)

System requirements

To view system requirements, go to <https://www.ibm.com/software/reports/compatibility/clarity/softwareReqsForProduct.html>.

Installing

Important: To install, you must be logged on to your computer with administrator privileges.

- [Installing from a downloaded file](#)
- [Silent installation](#)
- [Notes for installation](#)

Installing from a downloaded file

You must run the installer as administrator:

1. Double-click the file that you downloaded and extract all the files to some location on your computer.
2. Using Windows Explorer, browse to the location where you extracted the files.
3. Right-click *setup.exe* and choose Run as Administrator.
4. Follow the instructions that appear on the screen. See [Notes for installation](#) for any special instructions.

Silent installation

Silent mode enables an installation to run on its own without any interaction; installing silently can free system administrators from the task of monitoring each installation and providing input to prompts and dialog boxes. This method is especially useful when you are installing SPSS® Modeler on a number of different computers that have identical hardware.

Note: You must have administrator privileges to be able to run silent installations.

Windows - silent installation

You can complete a silent installation on Windows systems by using Microsoft Installer (MSI). Use *msiexec.exe* to install the MSI package.

The following options can be used:

Table 1. Silent installation options

Option	Description
/i	Specifies that the program is to install the product.
/i*v	Specifies verbose logging. For example, this form of log can be useful if you need to troubleshoot an installation.
/qn	Runs the installation without running the external user interface sequence.
/s	Specifies silent mode.
/v	Specifies that the Setup Program passes the parameter string to the call it makes to the MSI executable file (<i>msiexec.exe</i>). The following syntax requirements apply if you use this option: <ul style="list-style-type: none">• You must place a backslash (\) in front of any quotation marks (" ") that are within existing quotation marks.• Do not include a space between the /v option and its arguments.• Multiple parameters that are entered with the /v option must be separated with a space.• To create a log file, specify the directory and file name at the end of the command. The directory must exist before you start the silent installation.
/x	Specifies that the program is to uninstall the product.

An example of the MSI command is shown below.

Important: This command restarts the machine automatically. Ensure you save and close any open applications before running the command.

```
C:>msiexec.exe /i ModelerClient64.msi /qn /l*v  
c:\temp\Modeler_Silent_Install.log  
INSTALLDIR="C:\Program Files\IBM\SPSS\Modeler\19"  
LICENSETYPE="Network"  
LHOST="netlicense.mylocation.mycompany.com"
```

Note: Depending on your system, you might need to change the .msi file in the preceding example. The .msi versions for SPSS Modeler Client are shown in the following list.

- ModelerClient32.msi - 32-bit
- ModelerClient64.msi - 64-bit

If you are using a single license for your SPSS Modeler Client installation, remove the LICENSETYPE parameter and modify LHOST to = "no-net", as shown in the example below.

```
C:>msiexec.exe /i ModelerClient64.msi /qn /l*v  
c:\temp\Modeler_Silent_Install.log  
INSTALLDIR="C:\Program Files\IBM\SPSS\Modeler\19"  
LHOST="no-net"
```

When the installation is complete, ensure you run the License Authorization Wizard application to license SPSS Modeler Client.

Windows - silent uninstalling

The following text shows an example of the MSI command to silently uninstall the software:

```
C:>msiexec.exe /x ModelerClient64.msi /qn /norestart
```

Notes for installation

This section contains special instructions for this installation.

Older versions of IBM® SPSS® Modeler. On Windows, the installation will automatically find and upgrade 18.1 or later installations of IBM SPSS Modeler, so there's no need to uninstall the older version. If you have a pre-18.1 version installed, you'll need to uninstall it. On Linux and Mac, the installation does not automatically find 18.1 or later installations. But if you choose a directory that has an 18.1 or later installation present, the 18.1.1 or later installation will automatically upgrade it.

Licensing your product

You must run the License Authorization Wizard to license your product.

- [Using the license authorization wizard](#)
-

Using the license authorization wizard

Note: You might be prompted for administrator credentials. Without the correct credentials, you will not be able to run the License Authorization Wizard.

1. To launch the License Authorization Wizard, click License Product on the Welcome dialog or choose License Authorization Wizard in the Windows Start menu program group for IBM® SPSS® Modeler. You must run as administrator. Right-click the License Authorization Wizard shortcut and choose Run As Administrator.
2. Select Concurrent user license. When prompted, enter the license manager server name or IP address. This is the IP address or the name of the server on which the network license manager is running. If you have multiple addresses or names, separate them with a tilde (for example, server1~server2~server3). Contact your administrator if you do not have this information.
Note: Depending on your environment, you may need verify that TCP port 7 is open. The License Authorization Wizard needs to contact the license manager server one time on port 7 to verify it exists.

Invalid digital signature on installation

IBM® SPSS® Modeler products use IBM-issued certification for digital signing. In certain circumstances you may see the following error on trying to install SPSS Modeler products:

```
Error 1330. A file that is required cannot be installed because the cabinet file filename has an invalid digital signature...
```

All Windows users

You see this message if you try to install SPSS Modeler products on a machine that has no Internet connection and does not have the correct certificate installed. Use the following procedure to correct this problem.

1. Click OK to acknowledge the message.
2. Click Cancel to exit from the installer.
3. If the machine on which you want to install has no Internet connection, perform the next step on an Internet-connected machine and copy the .cer file to the machine where you want to install.

4. Go to <https://support.symantec.com>, search for VeriSign Class 3 Primary Certification Authority - G5 root certificate, and download it. Save it as a .cer file.
5. Double-click the .cer file.
6. On the General tab, click Install Certificate.
7. Follow the instructions in the Certificate Import Wizard, using the default options and clicking Finish at the end.
8. Retry the installation.

Configuring IBM SPSS Modeler to work with IBM SPSS Statistics

To enable IBM® SPSS® Modeler to use the Statistics Transform, Statistics Model, and Statistics Output nodes, you must have a copy of IBM SPSS Statistics installed and licensed on the computer where the stream is run.

If running IBM SPSS Modeler in local (standalone) mode, the licensed copy of IBM SPSS Statistics must be on the local computer.

When you have finished installing this copy of SPSS Modeler Client, you will also need to configure it to work with IBM SPSS Statistics. From the main client menu, choose:

Tools > Options > Helper Applications

and on the IBM SPSS Statistics tab, specify the location of the local IBM SPSS Statistics installation you want to use. For more information, see the *Source, Process and Output Nodes* guide or the online help for Helper Applications.

In addition, if running in distributed mode against a remote IBM SPSS Modeler Server, you also need to run a utility at the IBM SPSS Modeler Server host to create the **statistics.ini** file, which indicates to IBM SPSS Modeler Server the installation path for IBM SPSS Statistics Server. To do this, from the command prompt, change to the IBM SPSS Modeler Server bin directory and, for Windows, run:

```
statisticsutility -location=<statistics_installation_path>/bin
```

Alternatively, for UNIX, run:

```
./statisticsutility -location=<statistics_installation_path>/bin
```

Following is an example of what is placed in the **statistics.ini** file located in the IBM SPSS Modeler Server /bin directory after running the utility in IBM SPSS Modeler Server:

```
[LOCATION]
STATISTICS_PATH=C:\Program Files\IBM\SPSS\StatisticsServer\<version>
```

If you do not have a licensed copy of IBM SPSS Statistics on your local machine, you can still run the Statistics File node against a IBM SPSS Statistics server, but attempts to run other IBM SPSS Statistics nodes will display an error message.

Database Access

To read or write to a database, you must have an ODBC data source installed and configured for the relevant database, with read or write permissions as needed. The IBM SPSS Data Access Pack includes a set of ODBC drivers that can be used for this purpose, and these drivers are available from the download site. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

Supported ODBC drivers

For the latest information on which databases and ODBC drivers are supported and tested for use with IBM® SPSS® Modeler, see the product compatibility matrices on the corporate Support site (<http://www.ibm.com/support>).

Where to install drivers

Note: ODBC drivers must be installed and configured on each computer where processing may occur.

- If you are running IBM SPSS Modeler in local (standalone) mode, the drivers must be installed on the local computer.
- If you are running IBM SPSS Modeler in distributed mode against a remote IBM SPSS Modeler Server, the ODBC drivers need to be installed on the computer where IBM SPSS Modeler Server is installed.
- If you need to access the same data sources from both IBM SPSS Modeler and IBM SPSS Modeler Server, the ODBC drivers must be installed on both computers.
- If you are running IBM SPSS Modeler over Terminal Services, the ODBC drivers need to be installed on the Terminal Services server on which you have IBM SPSS Modeler installed.

Checking out/in a commuter license

Network licenses normally require that you are connected to the network to run IBM® SPSS® Modeler. If your administrator enabled commuter licenses, you can check out a commuter license to use the network license when you are not connected to the network. For example, you may want to run IBM SPSS Modeler on the train when you don't have a network connection. Before disconnecting from your network and catching the train, you could check out a commuter license for a limited amount of time. You will need to reconnect to the network and check the license back in before the time expires. Otherwise, IBM SPSS Modeler will stop working.

Network licenses are enabled and configured by your administrator. If you would like to use this feature and can't, check with your administrator.

Important: Even if you are able to run IBM SPSS Modeler because you are reconnected to the network, be sure to check the license back in. Doing so will allow other users to take advantage of the commuter license.

Check out a license

1. Choose Commuter License in the Windows Start menu program group for IBM SPSS Modeler.
2. Select the license that you want to check out.
3. In the Duration box, enter the number of days for which you want to check out the license. There is a limit that your administrator configures.
4. Click Check Out.

The commuter license will expire after the number of days specified by Duration. You can also manually check the license back in at any time.

You may receive a message in the following format:

Error while checkout with error code: <code>

Common codes are as follows.

Code	Meaning
77	All available licenses have been checked out.
1402	Attempt to check out license that has been reserved for another user.

Check in a license

1. Choose Commuter License in the Windows Start menu program group for IBM SPSS Modeler.
2. Select the license that you want to check in. License(s) that you checked out are indicated by a check mark.
3. Click Check In.

Applying fix packs

To ensure problem-free operation, keep your product at the latest fix pack level. Complete all of the necessary pre-installation and post-installation tasks as described in the fix pack instructions.

Uninstalling

To completely uninstall IBM® SPSS® Modeler:

1. Use the Windows Control Panel to remove IBM SPSS Modeler.

Installation instructions

The following instructions are for installing IBM® SPSS® Modeler version 18.4.0 using the license type authorized user license. This document is for users who are installing on their desktop computers.

- [System requirements](#)
- [Installing](#)
- [Licensing your product](#)
- [Configuring IBM SPSS Modeler to work with IBM SPSS Statistics](#)

- [Database Access](#)
 - [Applying fix packs](#)
 - [Uninstalling](#)
 - [Updating, modifying, and renewing IBM SPSS Modeler](#)
-

System requirements

To view system requirements, go to <https://www.ibm.com/software/reports/compatibility/clarity/softwareReqsForProduct.html>.

- [Authorization code](#)

You will also need your authorization code(s). In some cases, you might have multiple codes. You will need all of them.

Authorization code

You will also need your authorization code(s). In some cases, you might have multiple codes. You will need all of them.

You should have received separate instructions for obtaining your authorization code. If you cannot find your authorization code, contact Customer Service by visiting <https://www.ibm.com/products/spss-modeler/support>.

Installing

Important: To install, you must be logged on to your computer with administrator privileges.

- [Installing from a downloaded file](#)
 - [Silent installation](#)
 - [Notes for installation](#)
-

Installing from a downloaded file

You must run the installer as administrator:

1. Double-click the file that you downloaded and extract all the files to some location on your computer.
 2. Using Windows Explorer, browse to the location where you extracted the files.
 3. Right-click `setup.exe` and choose Run as Administrator.
 4. Follow the instructions that appear on the screen. See [Notes for installation](#) for any special instructions.
-

Silent installation

Silent mode enables an installation to run on its own without any interaction; installing silently can free system administrators from the task of monitoring each installation and providing input to prompts and dialog boxes. This method is especially useful when you are installing SPSS® Modeler on a number of different computers that have identical hardware.

Note: You must have administrator privileges to be able to run silent installations.

Windows - silent installation

You can complete a silent installation on Windows systems by using Microsoft Installer (MSI). Use `msiexec.exe` to install the MSI package.

The following options can be used:

Table 1. Silent installation options

Option	Description
<code>/i</code>	Specifies that the program is to install the product.
<code>/i*v</code>	Specifies verbose logging. For example, this form of log can be useful if you need to troubleshoot an installation.
<code>/qn</code>	Runs the installation without running the external user interface sequence.

Option	Description
/s	Specifies silent mode.
/v	Specifies that the Setup Program passes the parameter string to the call it makes to the MSI executable file (msiexec.exe). The following syntax requirements apply if you use this option: <ul style="list-style-type: none"> You must place a backslash (\) in front of any quotation marks (" ") that are within existing quotation marks. Do not include a space between the /v option and its arguments. Multiple parameters that are entered with the /v option must be separated with a space. To create a log file, specify the directory and file name at the end of the command. The directory must exist before you start the silent installation.
/x	Specifies that the program is to uninstall the product.

An example of the MSI command is shown below.

Important: This command restarts the machine automatically. Ensure you save and close any open applications before running the command.

```
C:>msiexec.exe /i ModelerClient64.msi /qn /l*v
c:\temp\Modeler_Silent_Install.log
INSTALLDIR="C:\Program Files\IBM\SPSS\Modeler\19"
LICENSETYPE="Network"
LSHOST="netlicense.mylocation.mycompany.com"
```

Note: Depending on your system, you might need to change the .msi file in the preceding example. The .msi versions for SPSS Modeler Client are shown in the following list.

- ModelerClient32.msi - 32-bit
- ModelerClient64.msi - 64-bit

If you are using a single license for your SPSS Modeler Client installation, remove the LICENSETYPE parameter and modify LSHOST to ="no-net", as shown in the example below.

```
C:>msiexec.exe /i ModelerClient64.msi /qn /l*v
c:\temp\Modeler_Silent_Install.log
INSTALLDIR="C:\Program Files\IBM\SPSS\Modeler\19"
LSHOST="no-net"
```

When the installation is complete, ensure you run the License Authorization Wizard application to license SPSS Modeler Client.

Windows - silent uninstalling

The following text shows an example of the MSI command to silently uninstall the software:

```
C:>msiexec.exe /x ModelerClient64.msi /qn /norestart
```

Notes for installation

This section contains special instructions for this installation.

Older versions of IBM® SPSS® Modeler. On Windows, the installation will automatically find and upgrade 18.1 or later installations of IBM SPSS Modeler, so there's no need to uninstall the older version. If you have a pre-18.1 version installed, you'll need to uninstall it. On Linux and Mac, the installation does not automatically find 18.1 or later installations. But if you choose a directory that has an 18.1 or later installation present, the 18.1.1 or later installation will automatically upgrade it.

Licensing your product

You must run the License Authorization Wizard to license your product.

Note: Licenses are tied to your computer's hardware with a **lock code**. If you replace your computer or its hardware, you will have a new lock code and will need to repeat the authorization process. This is also true if you re-image your computer. If you find out that you exceeded the allowable number of authorizations specified in the license agreement, go to <https://www.ibm.com/products/spss-modeler/support> to contact the Client Care team for assistance.

Important: The license is sensitive to time changes. If you must change the system time and then cannot run the product, contact the Client Care team for assistance by visiting <https://www.ibm.com/products/spss-modeler/support>.

- [Using the license authorization wizard](#)
- [Viewing your license](#)

Using the license authorization wizard

Note: You might be prompted for administrator credentials. Without the correct credentials, you will not be able to run the License Authorization Wizard.

1. To launch the License Authorization Wizard, click License Product on the Welcome dialog or choose License Authorization Wizard in the Windows Start menu program group for IBM® SPSS® Modeler. You must run as administrator. Right-click the License Authorization Wizard shortcut and choose Run As Administrator.
2. Select Authorized user license. When prompted, enter one or more authorization codes.
You should have received separate instructions for obtaining your authorization code. If you cannot find your authorization code, contact Customer Service by visiting <https://www.ibm.com/products/spss-modeler/support>.

The License Authorization Wizard sends your authorization code over the Internet to IBM Corp. and automatically retrieves your license. If your computer is behind a proxy, click Connect to the internet through a proxy server and enter the appropriate settings.

If the authorization process fails, you will be prompted to send an e-mail message. Choose whether you want to send the e-mail message through your desktop e-mail program or through a Web-based e-mail application.

- If you choose the desktop option, a new message with the appropriate information will be created automatically.
- If you choose the Web-based option, you must first create a new message in your Web-based e-mail program. Then copy the message text from the License Authorization Wizard and paste it into your e-mail application.

Send the e-mail message and respond to the prompt in the License Authorization Wizard. The e-mail message will be processed almost instantaneously. You can click Enter License Code(s) to enter any license code(s) that you receive. If you already closed the License Authorization Wizard, restart it and select Authorized user license. On the Enter Codes panel, add the license code that you received and click Next to complete the process.

Viewing your license

You can view the license by relaunching the License Authorization Wizard. The first panel displays the licensing information. Click Cancel when done, and click Yes when prompted about canceling.

Invalid digital signature on installation

IBM® SPSS® Modeler products use IBM-issued certification for digital signing. In certain circumstances you may see the following error on trying to install SPSS Modeler products:

Error 1330. A file that is required cannot be installed because the cabinet file filename has an invalid digital signature...

All Windows users

You see this message if you try to install SPSS Modeler products on a machine that has no Internet connection and does not have the correct certificate installed. Use the following procedure to correct this problem.

1. Click OK to acknowledge the message.
2. Click Cancel to exit from the installer.
3. If the machine on which you want to install has no Internet connection, perform the next step on an Internet-connected machine and copy the .cer file to the machine where you want to install.
4. Go to <https://support.symantec.com>, search for VeriSign Class 3 Primary Certification Authority - G5 root certificate, and download it. Save it as a .cer file.
5. Double-click the .cer file.
6. On the General tab, click Install Certificate.
7. Follow the instructions in the Certificate Import Wizard, using the default options and clicking Finish at the end.
8. Retry the installation.

Configuring IBM SPSS Modeler to work with IBM SPSS Statistics

To enable IBM® SPSS® Modeler to use the Statistics Transform, Statistics Model, and Statistics Output nodes, you must have a copy of IBM SPSS Statistics installed and licensed on the computer where the stream is run.

If running IBM SPSS Modeler in local (standalone) mode, the licensed copy of IBM SPSS Statistics must be on the local computer.

When you have finished installing this copy of SPSS Modeler Client, you will also need to configure it to work with IBM SPSS Statistics. From the main client menu, choose:

Tools > Options > Helper Applications

and on the IBM SPSS Statistics tab, specify the location of the local IBM SPSS Statistics installation you want to use. For more information, see the *Source, Process and Output Nodes* guide or the online help for Helper Applications.

In addition, if running in distributed mode against a remote IBM SPSS Modeler Server, you also need to run a utility at the IBM SPSS Modeler Server host to create the **statistics.ini** file, which indicates to IBM SPSS Modeler Server the installation path for IBM SPSS Statistics Server. To do this, from the command prompt, change to the IBM SPSS Modeler Server bin directory and, for Windows, run:

```
statisticsutility -location=<statistics_installation_path>/bin
```

Alternatively, for UNIX, run:

```
./statisticsutility -location=<statistics_installation_path>/bin
```

Following is an example of what is placed in the **statistics.ini** file located in the IBM SPSS Modeler Server /bin directory after running the utility in IBM SPSS Modeler Server:

```
[LOCATION]
STATISTICS_PATH=C:\Program Files\IBM\SPSS\StatisticsServer\<version>
```

If you do not have a licensed copy of IBM SPSS Statistics on your local machine, you can still run the Statistics File node against a IBM SPSS Statistics server, but attempts to run other IBM SPSS Statistics nodes will display an error message.

Database Access

To read or write to a database, you must have an ODBC data source installed and configured for the relevant database, with read or write permissions as needed. The IBM SPSS Data Access Pack includes a set of ODBC drivers that can be used for this purpose, and these drivers are available from the download site. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

Supported ODBC drivers

For the latest information on which databases and ODBC drivers are supported and tested for use with IBM® SPSS® Modeler, see the product compatibility matrices on the corporate Support site (<http://www.ibm.com/support>).

Where to install drivers

Note: ODBC drivers must be installed and configured on each computer where processing may occur.

- If you are running IBM SPSS Modeler in local (standalone) mode, the drivers must be installed on the local computer.
- If you are running IBM SPSS Modeler in distributed mode against a remote IBM SPSS Modeler Server, the ODBC drivers need to be installed on the computer where IBM SPSS Modeler Server is installed.
- If you need to access the same data sources from both IBM SPSS Modeler and IBM SPSS Modeler Server, the ODBC drivers must be installed on both computers.
- If you are running IBM SPSS Modeler over Terminal Services, the ODBC drivers need to be installed on the Terminal Services server on which you have IBM SPSS Modeler installed.

Applying fix packs

To ensure problem-free operation, keep your product at the latest fix pack level. Complete all of the necessary pre-installation and post-installation tasks as described in the fix pack instructions.

Uninstalling

To completely uninstall IBM® SPSS® Modeler:

1. Use the Windows Control Panel to remove IBM SPSS Modeler.

Updating, modifying, and renewing IBM® SPSS® Modeler

If you purchase additional options or renew your license, you will receive a new authorization code (or codes). .

Installation overview

The following instructions are for installing IBM® SPSS® Modeler version 18.4.0 on a Mac OS.

- [System requirements](#)
- [Installing](#)

System requirements

To view system requirements, go to <https://www.ibm.com/software/reports/compatibility/clarity/softwareReqsForProduct.html>.

- [License types](#)

License types

There are two types of licenses:

Authorized user license

If you are an individual who purchased IBM® SPSS® Modeler for yourself, you have an authorized user license. An authorized user license has an associated code that authorizes individual installations of the product.

If you are part of an organization, you might have an authorized user license for a specified number of users. The same authorization code is valid until the number of authorizations exceeds the specified number.

Concurrent user license

A concurrent user license is a "floating" license that can be used simultaneously (concurrently) by a specified number of users. Each installation of the product is not authorized. Instead, the product is authorized on a server machine through an application called the *license manager*. When the product starts up, it communicates with the server machine and checks if a license is currently available.

Installing

- [Installing from a downloaded file](#)
- [Notes for installation](#)

Installing from a downloaded file

1. Mount the installer disk image by double-clicking the file that you downloaded.
2. In the mounted disk image, double-click the installer file, and then follow the instructions that appear on the screen. See [Notes for installation](#) for any special instructions.

Notes for installation

This section contains special instructions for this installation.

Installer language. The first panel of the installer prompts for an installer language. By default, the language that matches your locale is selected. If you would like to display the installer in another language, select the language. Click OK when you are ready to proceed.

Older versions of SPSS® Modeler. The installation does not automatically overwrite earlier installations of SPSS Modeler. You have to uninstall older versions manually.

Authorized user license installation

- [Authorization code](#)
 - [Licensing your product](#)
 - [Using the license authorization wizard](#)
 - [Viewing your license](#)
 - [Updating, modifying, and renewing SPSS® Modeler](#)
-

Authorization code

You will need your authorization code(s). In some cases, you might have multiple codes. You will need all of them.

You should have received separate instructions for obtaining your authorization code. If you cannot find your authorization code, contact Customer Service by visiting <https://www.ibm.com/products/spss-modeler/support>.

Licensing your product

You must run the License Authorization Wizard to license your product.

Note: Licenses are tied to your computer's hardware with a lock code. If you replace your computer or its hardware, you will have a new lock code and will need to repeat the authorization process. This is also true if you re-image your computer. If you find out that you exceeded the allowable number of authorizations specified in the license agreement, go to <https://www.ibm.com/products/spss-modeler/support> to contact the Client Care team for assistance.

Important: The license is sensitive to time changes. If you must change the system time and then cannot run the product, contact the Client Care team for assistance by visiting <https://www.ibm.com/products/spss-modeler/support>.

Using the license authorization wizard

1. To launch the License Authorization Wizard, click License Product on the Welcome dialog or click the License Authorization Wizard icon in the SPSS® Modeler application folder.

2. Select Authorized user license. When prompted, enter one or more authorization codes.

You should have received separate instructions for obtaining your authorization code. If you cannot find your authorization code, contact Customer Service by visiting <https://www.ibm.com/products/spss-modeler/support>.

The License Authorization Wizard sends your authorization code over the Internet to IBM® Corp. and automatically retrieves your license. If your computer is behind a proxy, click Connect to the internet through a proxy server and enter the appropriate settings.

If the authorization process fails, you are prompted to send an e-mail message. Choose whether you want to send the e-mail message through your desktop e-mail program or through a Web-based e-mail application.

- If you choose the desktop option, a new message with the appropriate information is created automatically.
- If you choose the Web-based option, you must create a new message in your Web-based e-mail program. Then copy the message text from the License Authorization Wizard and paste it into your e-mail application.

Send the e-mail message and respond to the prompt in the License Authorization Wizard. The e-mail message is processed almost instantaneously. You can click Enter License Code(s) to enter any license code(s) that you receive. If you already closed the License Authorization Wizard, restart it and select Authorized user license. On the Enter Codes panel, add the license code that you received and click Next to complete the process.

Viewing your license

You can view the license by relaunching the License Authorization Wizard. The first panel displays the licensing information. Click Cancel when done, and, when prompted about canceling, click Yes.

Updating, modifying, and renewing SPSS® Modeler

If you purchase additional options or renew your license, you will receive a new authorization code (or codes). For instructions on using the authorization code(s), see [Licensing your product](#).

Concurrent user license installation

- [Using the license authorization wizard](#)
- [Checking out/in a commuter license](#)

Using the license authorization wizard

1. To launch the License Authorization Wizard, click License Product on the Welcome dialog or click the License Authorization Wizard icon in the SPSS® Modeler application folder.
2. To launch the License Authorization Wizard, run *law.exe*, which can be found in the license manager installation directory.
3. Select Concurrent user license. When prompted, enter the license manager server name or IP address. This is the IP address or the name of the server on which the network license manager is running. If you have multiple addresses or names, separate them with a tilde (for example, *server1~server2~server3*). Contact your administrator if you do not have this information.

Checking out/in a commuter license

Network licenses normally require that you are connected to the network to run SPSS® Modeler. If your administrator enabled commuter licenses, you can check out a commuter license to use the network license when you are not connected to the network. For example, you may want to run SPSS Modeler on the train when you don't have a network connection. Before disconnecting from your network and catching the train, you could check out a commuter license for a limited amount of time. You will need to reconnect to the network and check the license back in before the time expires. Otherwise, SPSS Modeler will stop working.

Network licenses are enabled and configured by your administrator. If you would like to use this feature and can't, check with your administrator.

Important: Even if you are able to run SPSS Modeler because you are reconnected to the network, be sure to check the license back in. Doing so will allow other users to take advantage of the commuter license.

Check out a license

1. Double-click *Commuter Utility* in the installation directory.
2. Select the license that you want to check out.
3. In the Duration box, enter the number of days for which you want to check out the license. There is a limit that your administrator configures.
4. Click Check Out.

The commuter license will expire after the number of days specified by Duration. You can also manually check the license back in at any time.

You may receive a message in the following format:

Error while checkout with error code: <code>

Common codes are as follows.

Code	Meaning
77	All available licenses have been checked out.
1402	Attempt to check out license that has been reserved for another user.

Check in a license

1. Double-click *Commuter Utility* in the installation directory.
2. Select the license that you want to check in. License(s) that you checked out are indicated by a check mark.
3. Click Check In.

After installation

Depending on the components you have for SPSS® Modeler, you may have to carry out further configuration after installing the main software. For example, this might be to connect to a database, or to use data that is compatible with IBM® SPSS Statistics.

- [Configuring IBM SPSS Modeler to work with IBM SPSS Statistics](#)
 - [Database Access](#)
 - [Applying fix packs](#)
 - [Uninstalling](#)
-

Configuring IBM SPSS Modeler to work with IBM SPSS Statistics

To enable IBM® SPSS® Modeler to use the Statistics Transform, Statistics Model, and Statistics Output nodes, you must have a copy of IBM SPSS Statistics installed and licensed on the computer where the stream is run.

If running IBM SPSS Modeler in local (standalone) mode, the licensed copy of IBM SPSS Statistics must be on the local computer.

When you have finished installing this copy of SPSS Modeler Client, you will also need to configure it to work with IBM SPSS Statistics. From the main client menu, choose:

Tools > Options > Helper Applications

and on the IBM SPSS Statistics tab, specify the location of the local IBM SPSS Statistics installation you want to use. For more information, see the *Source, Process and Output Nodes* guide or the online help for Helper Applications.

In addition, if running in distributed mode against a remote IBM SPSS Modeler Server, you also need to run a utility at the IBM SPSS Modeler Server host to create the **statistics.ini** file, which indicates to IBM SPSS Modeler Server the installation path for IBM SPSS Statistics Server. To do this, from the command prompt, change to the IBM SPSS Modeler Server bin directory and, for Windows, run:

```
statisticsutility -location=<statistics_installation_path>/bin
```

Alternatively, for UNIX, run:

```
./statisticsutility -location=<statistics_installation_path>/bin
```

Following is an example of what is placed in the **statistics.ini** file located in the IBM SPSS Modeler Server /bin directory after running the utility in IBM SPSS Modeler Server:

```
[LOCATION]
STATISTICS_PATH=C:\Program Files\IBM\SPSS\StatisticsServer\<version>
```

If you do not have a licensed copy of IBM SPSS Statistics on your local machine, you can still run the Statistics File node against a IBM SPSS Statistics server, but attempts to run other IBM SPSS Statistics nodes will display an error message.

Database Access

To read or write to a database, you must have an ODBC data source installed and configured for the relevant database, with read or write permissions as needed. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

Supported ODBC drivers

For the latest information on which databases and ODBC drivers are supported and tested for use with IBM® SPSS® Modeler, see the product compatibility matrices on the corporate Support site (<http://www.ibm.com/support>).

Where to install drivers

Note: ODBC drivers must be installed and configured on each computer where processing may occur.

- If you are running IBM SPSS Modeler in local (standalone) mode, the drivers must be installed on the local computer.
- If you are running IBM SPSS Modeler in distributed mode against a remote IBM SPSS Modeler Server, the ODBC drivers need to be installed on the computer where IBM SPSS Modeler Server is installed. For IBM SPSS Modeler Server on UNIX systems, see also "Configuring ODBC drivers on UNIX systems" later in this section.
- If you need to access the same data sources from both IBM SPSS Modeler and IBM SPSS Modeler Server, the ODBC drivers must be installed on both computers.
- If you are running IBM SPSS Modeler over Terminal Services, the ODBC drivers need to be installed on the Terminal Services server on which you have IBM SPSS Modeler installed.

Applying fix packs

To ensure problem-free operation, keep your product at the latest fix pack level. Complete all of the necessary pre-installation and post-installation tasks as described in the fix pack instructions.

Uninstalling

The IBM SPSS modeler installation files are hidden under the Home folder. Before you uninstall IBM SPSS modeler, enable the view of hidden folders in your Home folder.

Complete the following steps to uninstall IBM SPSS Modeler on Mac OS.

1. Move the installation folder to Bin.
The default installation folder is /Applications/IBM/SPSS/Modeler/18.4.0.
2. Move the file Library/Preferences/com.ibm.spss.plist from the Home folder to Bin.
3. Move the folder ./IBM/SPSS/Modeler/18.4.0 (located in Home folder) to Bin.
4. Remove any extension commands that you installed.
The extension commands are in the ext folder. Move the folder ./IBM/SPSS/Modeler/18.4.0/ext to Bin
5. Empty Bin.

Administrator's guide

The following instructions are for administrators at sites with the license type concurrent license for IBM® SPSS® Modeler 18.4.0. This license allows you to install IBM SPSS Modeler on any number of computers. However, only the number of users for which you purchased the license can run the application concurrently.

- [Before you start](#)
- [Installing the concurrent license manager](#)
- [Licensing your product](#)
- [Testing the license manager](#)
- [Installing the product on the local desktop computers](#)
- [Administering the concurrent license](#)

Before you start

You will need your authorization code. The authorization code enables you to get a license for the concurrent license manager. The **concurrent license manager** controls your concurrent license and allows end users to run IBM® SPSS® Modeler.

You should have received separate instructions for obtaining your authorization code. If you cannot find your authorization code, contact Customer Service by visiting <https://www.ibm.com/products/spss-modeler/support>.

To ensure that you set up the concurrent license correctly, follow these steps:

1. **Install the concurrent license manager.** The concurrent license manager is the utility that will serve up the concurrent licenses to end users. You can install the license manager on any computer on your network. This is typically a server to which desktop computers can connect. See the topic [Installing the concurrent license manager](#) for more information.
2. **Lic平 your product.** This action will give the license manager the required information for serving up the licenses. See the topic [Licensing your product](#) for more information.
3. **Test the concurrent license manager.** You should test license manager to make sure it is serving up licenses. See the topic [Testing the license manager](#) for more information.
4. **Install IBM SPSS Modeler on the local desktop computers.** You or your end users will complete the full installation on the desktop computers. During an installation, an end user can specify the computer on which the license manager is running. When an end user tries to launch IBM SPSS Modeler, the product communicates with this license manager to get a license. If a license is available, IBM SPSS Modeler launches. See the topic [Installing the product on the local desktop computers](#) for more information.

No administration steps are required for the installation, but if any problems arise, see [Administering the concurrent license](#).

- [Citrix and Terminal Services](#)
- [Mixed licensing](#)

Citrix and Terminal Services

You can install and publish the IBM® SPSS® Modeler application on your Citrix or Terminal Services Server the same way that you install and publish other Windows applications.

Mixed licensing

IBM® SPSS® Modeler has limited support for a mixed licensing environment (a combination of local authorized user licenses and concurrent licenses). In a typical environment, all licenses are handled locally or through the concurrent license manager. For example, if the main product is licensed through the concurrent license manager, options are also licensed through the concurrent license manager.

The only option for mixed licensing is to license the main product through the concurrent license manager and to license the options on the end user's computer with *licenseactivator*. To set up this type of mixed licensing, install IBM SPSS Modeler with a concurrent license. Then use *licenseactivator* on the end user's computer to license the options. For more information about *licenseactivator*, see [Using licenseactivator to install a license automatically](#).

IBM SPSS Modeler does not support the opposite scenario to license the main product locally and to license the options through the concurrent license manager.

Installing the concurrent license manager

Before end users install IBM® SPSS® Modeler on their desktop computers, you must install the concurrent license manager. This utility is a continuously executing service or daemon that you will typically install on one computer on your network. (You also have the option of installing on multiple computers, in which case you would set up redundant license managers. For more information, see the topic [Setting up redundant license servers](#).) You do not install the concurrent license manager on the end users' desktop computers.

Whenever an end user starts a concurrent-licensed IBM SPSS application, the application will request a license from the license manager. This utility will deliver licenses up to a fixed number of simultaneous end users, determined by the license that you have purchased. After this number has been met, any further requests for licenses will be refused. If you find that end users are often being refused licenses, you can contact your sales representative to purchase a license that will allow more simultaneous users.

Note: The license manager will record the number of active end-user sessions and information about each session in a log file, which you can use to troubleshoot connection problems. See the topic [Obtaining log information](#) for more information.

Multiple operating systems

The platform on which the concurrent license manager is running does not have to match the platform on which the client is running. For example, a Linux license manager can serve up licenses for Windows clients.

Administering the license manager

You administer the license manager using the license manager administrator, which is available only for Windows. Therefore, you need at least one Windows machine on which to install the license manager administrator.

System requirements

The computer on which you install the concurrent license manager must meet the minimum operating system requirements. The license manager does not require a server class machine.

Important: If a firewall is running on the computer on which the license manager is installed, you must open port 5093 for UDP. Otherwise, client computers will not be able to retrieve a license from the license manager. Furthermore, if you are using redundant license manager machines, you must open port 5099 for UDP to enable the license manager machines to communicate with each other.

- [Upgrading the license manager](#)
- [Installing the license manager on Windows](#)
- [Installing the license manager on Mac OS](#)
- [Installing the license manager on non-Windows systems](#)
- [Installing the license manager administrator](#)

Upgrading the license manager

If you have a previous version of the license manager installed, you must perform the following steps:

1. Go to the directory in which you installed the old license manager.
2. Copy the *lservrc* file. On Windows, look in the *winnt* subdirectory.
3. Save the *lservrc* file to a safe location.
4. If any users have checked out computer licenses, make sure those users check the licenses back in. See the topic [Configuring computer licenses](#) for information about obtaining a list of checked out licenses.
5. Shutdown the license manager. See the topic [Starting and stopping the license manager](#) for more information.
6. Uninstall the old license manager. See the topic [Uninstalling the license manager](#) for more information.
7. Install the new license manager. For information about installing on Windows, see the topic [Installing the license manager on Windows](#). For information about installing on UNIX/Linux, see the topic [Installing the license manager on non-Windows systems](#).
8. Copy the saved *lservrc* file to the directory in which you installed the new license manager or the *winnt* subdirectory of the installation directory on Windows. If you accepted the default location on Windows, check the C:\Program Files (x86)\Common Files\SafeNet Sentinel\Sentinel RMS License Manager\WinNT folder.

Installing the license manager on Windows

Note: You must launch the installer as administrator. When instructed to launch an installer file, right-click the file and choose Run As Administrator.

Note: Ensure that the machine on which you are installing has Java installed.

1. For your downloaded eImage file, use an archive utility such as WinZip to extract all the files in the appropriate eImage.
2. Extract the contents of the archive that contains the license manager and tools.
3. Run setup.exe from the extracted manager directory and follow the instructions that appear on the screen.
4. To install the license manager administrator on the same machine as the license manager, run setup.exe from the extracted tools directory and follow the instructions that appear on the screen. You also have the option of installing the license manager administrator on another Windows machine if you want to administer the license manager remotely.

Installing the license manager on Mac OS

1. Insert the concurrent licensing tools DVD into the DVD drive of the network computer on which you want to run the license manager. Look for the archive that contains the license manager and tools for your operating system.

-or-

If you downloaded an eImage file, go to the location where you downloaded the file.

2. Extract the contents of the archive that contains the license manager and tools to the location where you want to install the license manager.
3. Install the license manager administrator on a Windows machine. See the topic [Installing the license manager administrator](#) for more information.
4. Refer to [Starting and stopping the license manager](#) for information about starting the license manager.

Installing the license manager on non-Windows systems

1. Go to the location where you downloaded your eImage file.
2. Extract the contents of the archive that contains the license manager and tools to the location where you want to install the license manager.
3. Install the license manager administrator on a Windows machine. See the topic [Installing the license manager administrator](#) for more information.
4. Refer to [Starting and stopping the license manager](#) for information about starting the license manager.

Installing the license manager administrator

The Windows-only license manager administrator is used to administer the license manager. If you installed the license manager on a non-Windows machine, you must install the license manager administrator on a Windows machine.

1. For your downloaded eImage file, use an archive utility such as WinZip to extract all the files in the appropriate eImage.
2. Extract the contents of the archive that contains the license manager and tools.
3. Run setup.exe from the extracted tools directory and follow the instructions that appear on the screen.

Licensing your product

On non-Windows operating systems, you must install the license from the command prompt after installing the license manager.

Note: Licenses are tied to the network computer's physical or virtual hardware with a **lock code**. If you replace the network computer or its hardware, you will have a new lock code and will need to contact your sales representative to obtain a new authorization code. If you are installing on a virtual machine, you need to ensure that you select a lock code that does not change on restart. For more information, see [Installing a license in a virtual environment](#).

Important: The license is sensitive to time changes. If you must change the system time and then cannot run the product, contact the Client Care team for assistance by visiting <https://www.ibm.com/products/spss-modeler/support>.

- [Installing a license in a virtual environment](#)
- [Installing a license from the command prompt](#)
- [Adding a license](#)
- [Viewing your license](#)

Installing a license in a virtual environment

If you installed the concurrent license manager in a virtual environment, there are special instructions for licensing. On a virtual machine the hardware is virtual, and the locking code that ties the license manager to the license might change when the virtual machine is restarted. To ensure the license manager works correctly, you need to find a locking code that does not change when the virtual machine is restarted. When you find a stable locking code, you will use it to license the license manager.

Important:

If you choose a locking code that does change on reboot, the license manager will stop working. The IBM® SPSS® application will not be able to retrieve a license and will fail to start.

Checking the locking code

1. Open a command prompt.
2. Change to the following directory.
 - Windows. The license manager administrator installation directory. If you accepted the default location during installation, the license manager administrator installation directory is C:\Program Files (x86)\Common Files\SafeNet Sentinel\SafeNet License Manager\WinNT.
 - Other platforms. The license manager installation directory.
3. At the command prompt, type `echo!d` (Windows) or `./echo!d` (other platforms).

You will see something like the following in the output:

Locking Code 1 : 4-12A1B

The number that appears immediately before the hyphen (-) is the locking code criteria. The locking code criteria is a number that represents the virtual hardware that is used for the locking code (in this example, it is 4, which represents the OS volume serial ID). The number after the hyphen is the locking code itself (in this example, it is 12A1B).

Following are the possible locking code criteria.

Locking code criteria	Virtual hardware
2	IP address
4	OS volume serial ID
8	Hostname
10	Ethernet card

Confirming that the locking code is stable

1. After checking the locking code, restart the virtual machine.

2. Check the locking code again (see [Checking the locking code](#)).
 - If the locking code **doesn't change**, reboot and check a few more times. If the locking code is stable, you are ready to license (see [Licensing the license manager with the new locking code](#)).
 - If the locking code **does change**, you need to update the locking code (see [Updating the locking code](#)).

Updating the locking code

1. In a text editor, open echoid.dat, which you can find in the following directory.
 - Windows. The license manager administrator installation directory. If you accepted the default location during installation, the license manager administrator installation directory is C:\Program Files (x86)\Common Files\SafeNet Sentinel\Sentinel RMS License Manager\WinNT.
 - Other platforms. The license manager installation directory.
2. You will see a single hexadecimal number that represents the current locking code criteria. Change this number to one of the acceptable locking code criteria in hexadecimal format.

Locking code criteria in hexadecimal format	Virtual hardware
0x002	IP address
0x004	OS volume serial ID
0x008	Hostname
0x010	Ethernet card

Licensing the license manager with the new locking code

After you find and update to a stable locking code, there are no further licensing steps that are particular to virtual environments. Use the command prompt to complete installation of the license.

Installing a license from the command prompt

You have two options for installing from the command prompt. You can use *licenseactivator* to get a license from the Internet automatically, or you can use *echoid* to get a license manually.

Note: When using silent installation for redundant license servers you must restart the machine when the installation is finished.

- [Using licenseactivator to install a license automatically](#)
- [Installing a license manually](#)

Using licenseactivator to install a license automatically

The computer on which you are installing the license must be connected to the Internet. If it isn't, install the license manually. See the topic [Installing a license manually](#) for more information.

1. Log in as the user who installed the license manager.
2. Open a command prompt and change directories to the license manager administrator installation directory. This is the directory in which you installed the *license manager administrator*, not the directory in which you installed IBM® SPSS® Modeler. If you accepted the default location on Windows, check the C:\Program Files (x86)\Common Files\SafeNet Sentinel\Sentinel RMS License Manager\WinNT folder.
3. You typically have an authorization code. In the simplest case, you enter the following at the command prompt/terminal window. See below for more details about the command prompt usage.

Windows: licenseactivator <auth-code>

UNIX/Linux/MacOS: ./licenseactivator <auth-code>

where <auth-code> is your authorization code.

You should see a message that the license was added successfully. If it wasn't, note the error code and try installing the license manually. See the topic [Installing a license manually](#) for more information.

When you use *licenseactivator*, it licenses the product and writes a log file to its directory. The name of the log file is *licenseactivator_<month>_<day>_<year>.log*. If any errors occur, you can check the log file for more information. This information is also useful if you contact IBM Corp. for support.

Using licenseactivator with Authorization Codes

licenseactivator is typically used with one or more authorization codes that you received when you purchased the product. Enter all of the text on one line.

Windows:

```
licenseactivator authcode1[:authcode2:...:authcodeN] [PROXYHOST=proxy-hostname] [PROXYPORt=proxy-port-number]
[PROXYUSER=proxy-userid] [PROXPASS=proxy-password]
```

UNIX/Linux/MacOS:

```
./licenseactivator authcode1[:authcode2:...:authcodeN] [PROXYHOST=proxy-hostname] [PROXYPORt=proxy-port-number]
[PROXYUSER=proxy-userid] [PROXPASS=proxy-password]
```

- Multiple authorization codes are separated by colons (:).
- The proxy settings are optional, but you may need them if your computer is behind a proxy. Which proxy settings are needed depend on your specific proxy configuration. You might need all of them.

PROXYHOST

The server name or IP address of the proxy host

PROXYPORt

The port number for connecting to the Internet through the proxy

PROXYUSER

If required, the user ID for the proxy

PROXPASS

If required, the password associated with the user ID

Using licenseactivator with License Codes

In less common scenarios, IBM Corp. may have sent you a *license*.

Windows:

```
licenseactivator licensocode[:licensocode2:...:licensocodeN]
```

UNIX/Linux/MacOS:

```
./licenseactivator licensocode[:licensocode2:...:licensocodeN]
```

- Multiple license codes are separated by colons (:).
- When using license codes, *licenseactivator* does not connect to the Internet, so you do not need to specify proxy information.

Installing a license manually

1. Log in as the user who installed the license manager.
2. Open a command prompt and change directories to the license manager administrator installation directory. Note that this is the directory in which you installed the *license manager administrator*, not the directory in which you installed IBM® SPSS® Modeler. If you accepted the default location on Windows, check the C:\Program Files (x86)\Common Files\SafeNet Sentinel\Sentinel RMS License Manager\WinNT folder.
3. Get the lock code for the server machine. At the command prompt, type `echoid` (Windows) or `./echoid` (other platforms).
4. Send the lock code and your authorization code to IBM Corp. by calling your local office or sending an e-mail message to `spsscs@us.ibm.com`. IBM Corp. will then provide a license code or a file containing a license code.
5. Use *licenseactivator* to enter the license code or codes.

Adding a license

You may want to add a license at a later time. The process for adding a license is the same as installing the original license.

Viewing your license

You can view your concurrent license (including the number of users) in the WlmAdmin application. For information about the WlmAdmin application and specifics about viewing the license, see [Administering the concurrent license](#).

Testing the license manager

To make sure the license manager is serving up licenses correctly, you should test it.

1. If you haven't installed the license manager administrator on another machine, install it on a Windows machine that is *not* running the license manager you want to test. See the topic [Installing the license manager administrator](#) for more information.
2. Install another license manager on a Windows machine that is *not* running the license manager you want to test. See the topic [Installing the license manager on Windows](#) for more information.
3. Start the WlmAdmin application. See the topic [Starting the WlmAdmin Application](#) for more information.
4. Add the remote license manager server that you want to test. See the topic [Adding a server](#) for more information.
5. View the licenses on the remote server. See the topic [Viewing details about a license](#) for more information.

If you are able to view the license, the license manager is ready for local desktop computers to connect to it. You can proceed with installing the product on local desktop computers. If you are not able to view the license, review the previous steps to ensure the license manager was installed correctly.

Installing the product on the local desktop computers

You have two options for installing the full product locally on an end user's computer. You can manually install on each computer, or you can use an application like Systems Management Server (SMS) to push the installation to the computers running Windows.

To manually install on a local desktop

1. **Make the installation media available.** Download the eImage for the product and extract the files to a shared network drive.
2. **Copy the installation instructions and prepare licensing information.** Make as many copies of the product installation instructions as you need. The installation instructions are available from the download site. Look for the instructions that correspond to your license type. After installation, the end user must enter the IP address or the name of the network computer on which the concurrent license manager is running. Fill out this information in the space provided at the beginning of the instructions before copying them.
3. **Distribute the installation materials to end users.** Distribute the downloaded file (or network location), the installation instructions, and the licensing information to end users who can manually install on each computer as needed.

To push to the local desktops running Windows

Because IBM® SPSS® Modeler installations are compatible with Microsoft Windows Installer (MSI), you can push an installation to the end-user desktop computers.

- [Pushing an installation to Windows computers](#)

Pushing an installation to Windows computers

Pushing an installation is a method for remotely distributing software to any number of end users without any user intervention. You can push the full installation of IBM® SPSS® Modeler to the end-user desktop computers running Windows. The technology that you are using for pushing the installation must support the MSI 3.0 engine or higher.

- [Uninstalling a Previous Version](#)
- [Properties for push installations](#)
- [MSI files](#)
- [Command line example](#)
- [Using SMS to push the installation](#)
- [Using Group Policy or related technology to push the installation](#)
- [Pushing an uninstallation](#)

Uninstalling a Previous Version

If you are going to push to the same directory in which a previous version of IBM® SPSS® Modeler was installed, you need to uninstall the old version. You must manually uninstall any IBM SPSS Modeler versions prior to 11.0 since push installations were not available for those versions. You can push the uninstallation as you push an installation. See the topic [Pushing an uninstallation](#) for more information.

Properties for push installations

Following are the properties that you can use for push installations. All properties are case sensitive. Values must be quoted if they contain spaces.

Table 1. Properties for push installations

Property	Description	Valid values	Default (If Applicable)
INSTALLDIR	The directory where IBM® SPSS® Modeler should be installed on the end user's desktop computer. This property is optional. If it is excluded, the default is C:\Program Files\IBM\SPSS\Modeler\18.4.0.	A valid path such as C:\Program Files\IBM\SPSS\Modeler\18.4.0.	C:\Program Files\IBM\SPSS\Modeler\18.4.0
LICENSETYPE	The license type. The value is case sensitive.	Network	
LSHOST	The IP addresses or the names of the network computer or computers on which the concurrent license manager is running.	One or more valid IP addresses or network computer names. Multiple addresses or names are separated by tildes (for example, server1~server2~server3).	

MSI files

Extract the contents of your downloaded eImage to access the MSI file. The file is located under the *modeler|<architecture>* directory, where <architecture> is 32bit or 64bit.

Command line example

Following is a command line that you could use to push a product installation. Enter all of the text on one line.

```
MsiExec.exe /i "modelerclient.msi" /qn /L*v logfile.txt  
INSTALLDIR="C:\Program Files\IBM\SPSS\Modeler\18.4.0" LICENSETYPE="Network" LSHOST="mylicserver"
```

Using SMS to push the installation

The basic steps for using Systems Management Servers (SMS) to push IBM® SPSS® Modeler are:

1. Extract the contents of your downloaded eImage and copy the appropriate subdirectory under the *modeler|<architecture>* directory to a directory on a network computer.
2. Edit the .pdf file located in the copied directory. Using a text editor, modify the value of CommandLine by adding the appropriate properties. For a list of the available properties, refer to [Properties for push installations](#). Make sure to specify the correct MSI file in the command line.
3. Create a package from the .pdf file and distribute the package to the end-user desktop machines.

Using Group Policy or related technology to push the installation

1. Extract the contents of your downloaded eImage and copy the appropriate subdirectory under the *modeler|<architecture>* directory to a directory on a network computer.
2. Using an application like ORCA, edit the Properties table in the appropriate file under the copied folder. ORCA is part of the Windows 2003 Server SDK, which you can find at <http://www.microsoft.com/downloads> by searching for the SDK. For a list of the properties that you can add to the Properties table, refer to [Properties for push installations](#). Make sure to use the correct MSI file.
3. Create a package using the edited file and distribute the package to the end-user desktop computers.

Pushing an uninstallation

Note: When you push the uninstall command, the end user loses customizations. If specific users require customizations, you can exclude those users from the distribution and ask them to install the product manually.

If you push an installation of a later version of IBM® SPSS® Modeler, you may want to uninstall first. You can do this silently by pushing the following command. Enter all of the text on one line.

```
MsiExec.exe /X{} /qn /L*v logfile.txt  
ALLUSERS=1 REMOVE="ALL"
```

The product code for a specific version is in the *setup.ini* file within each version's installed folders.

Administering the concurrent license

The license manager maintains your concurrent license. To administer the license manager itself and to view information about the concurrent licenses that it is maintaining, you can use the WlmAdmin application, which is the main user interface for the license manager administrator. If you are administering a license manager on a non-Windows machine or a remote Windows machine, install the license manager administrator on a separate Windows machine. See the topic [Installing the license manager administrator](#) for more information.

Note: If you need additional administration information, refer to the SafeNet documentation, which is installed with the license manager administrator. This documentation is in the *help\Content* directory in the license manager administration installation directory (for example, C:\Program Files (x86)\Common Files\SafeNet Sentinel\Sentinel RMS License Manager\help\Content).

- [Starting the WlmAdmin Application](#)
- [Adding a server](#)
- [Obtaining log information](#)
- [Viewing details about a license](#)
- [Setting up redundant license servers](#)
- [Configuring computer licenses](#)
- [Configuring license reservations](#)
- [Starting and stopping the license manager](#)
- [Uninstalling the license manager](#)
- [Uninstalling the license manager administrator](#)
- [Troubleshooting Desktop Computers](#)

Starting the WlmAdmin Application

From the Windows Start menu, choose:

[All] Programs>IBM>SPSS License Tools><version>>Sentinel RMS Server Administration

Adding a server

Before you can administer a license manager, you need to add its server to the WlmAdmin application. You have two options for adding the server.

To manually add a server

1. From the WlmAdmin application menus, choose:
Edit>Defined Server List
2. In the Defined Server List dialog, enter the name or IP address of the server on which the license manager is running.
3. Click Add.
4. Click OK.

The server now appears in the Defined Servers lists in the left pane of the WlmAdmin application.

To view a list of servers on the subnet

1. In the left pane of the WlmAdmin application, click the + sign next to Subnet Servers.

A list of license manager servers on your subnet appears. If you can't find a specific server with this method, manually add it as described above.

Obtaining log information

If end users have difficulty checking out licenses, the log files may contain useful information. You can use the `LSERVOPTS` environment variable and the `-f <trace-log-file>` and `-l <usage-log-file>` options to specify that log files should be created. For more information about this environment variable and its options, refer to the SafeNet documentation in the `Content` directory in the license manager administrator installation directory.

Viewing details about a license

You can view details about licenses that you added either manually or through the License Authorization Wizard.

1. In the left pane of the WlmAdmin application, click the + sign next to the license manager server to see the license(s).
2. Click the name of the license. The right pane displays details about the license. Codes are used to identify the licenses. The first part of the code indicates the feature. The second part indicates the version.

To see the names associated with the feature codes

1. Using a command prompt, change to the directory in which the license manager administrator is installed.
2. Type `lmshowlic <server>` (Windows) or `./lmshowlic <server>` (other operating systems), where `<server>` is the name or IP address of the server on which the license manager is running.

The output lists all the features available on the server, grouped by product and version.

Setting up redundant license servers

You can set up multiple, redundant license servers that support the same users. Redundant servers can help prevent any interruption that may occur when a server crashes. Another redundant server can take over the management of the license when the first server crashes.

You will need a special license code to enable the redundancy feature, as described in the following steps. For assistance with creating a redundant license key or any other licensing issue, contact IBM Support by phone or e-mail. You can find contact information at <http://www.ibm.com/planetwide>.

There must be an odd number of servers (at least three of them), and a majority must be running at the same time. For example, if there are three redundant license servers, two of them must be running.

To prepare each redundant license server

1. Install the license manager. See the topic [Installing the concurrent license manager](#) for more information.
2. Using a command prompt, change to the directory in which you installed the license manager.
3. Get the lock code for each server machine. At the command prompt, type `echoid` (Windows) or `./echoid` (other operating systems).
4. Write down the lock code. You will need the lock code for the next steps.
5. Repeat these steps for each redundant license server.

To activate the redundant licenses

1. Go to the IBM SPSS License Key Center (<https://spss.subscribenet.com/control/ibmp/login>).
2. Create a concurrent authorization code.
3. After the concurrent authorization code is created, click the code and then scroll down until you see the fields for Lock Code. You will now have the ability to add multiple lock codes to the license key.
4. Using the lock codes from the previous steps, enter the lock codes into the appropriate fields.
5. Click Submit.

To set up the redundant license server pool

1. If a license manager is running on any of the redundant license servers, stop the license manager on each computer.
2. From the WlmAdmin application menus, choose:
`Edit > Redundant License File`

This action opens the `WrlfTool` application.

3. From the WrlfTool application menus, choose:
File > New
4. For each redundant license server, click Add to specify the hostname and IP address of each server.
5. Change the order of the servers to indicate the order in which the redundant license servers are used. The first one in the list is the primary server.
6. Click Add License to add the license(s) that you received from Customer Service or your local office. If you received multiple licenses, be sure to add every one.
7. Click OK.
8. Click Done when you are finished.

To save the redundant license file

1. From the WrlfTool application menus, choose:
File > Save As
2. Save the redundant license file (*lsvrflf*) to an easily accessible location. You will need to copy the file in the next steps.

To configure the redundant license servers

1. Copy the redundant license file (*lsvrflf*) to the *winnt* subfolder of the license manager installation directory on Windows. If you accepted the default location, check the C:\Program Files (x86)\Common Files\SafeNet Sentinel\Sentinel RMS License Manager folder. On other operating systems, copy the file directly to the license manager installation directory. There must be at least three redundant license servers.
2. Start the license manager on each redundant license server.

To configure end-user computers

When the end user installs the product, the user specifies all redundant servers, with the server names or IP addresses separated by a tilde (for example, *server1~server2~server3*). The setup program then adds the necessary configuration information to the end user's computer. If the product is already installed on the desktop computers, you can perform the following manual steps to add the configuration information. You can also push an uninstall followed by a new install that defines all the servers. Refer to [Pushing an installation to Windows computers](#) for information about pushing installations.

1. Using a text editor, open *spssprod.inf*. On Windows, this file is located in the product installation directory on the desktop computer.
2. Change the value of **DAEMONHOST** to the server names or IP addresses separated by a tilde (~). For example:

```
DAEMONHOST=server1~server2~server3
```

3. Save *spssprod.inf*.

Configuring commuter licenses

Commuter licenses allow your end users to check out licenses from the license manager, so that they can use the license when not connected to the network. Commuter licenses are enabled by default. Instructions for actually checking out the commuter license appear in the end user installation instructions.

You can restrict the percentage of licenses that are enabled for commuting on the license manager server. It's a good idea to restrict commuter licenses to prevent all the licenses (tokens) from being used up by commuters. After the specified percentage of licenses have been used by commuters, no more will be available until the commuter licenses expire or are checked back in. You can also configure the maximum duration for which an end user can check out a license. The default maximum duration is 3 days.

Important: If you are using redundant license servers with commuter licenses, only the primary license server allows users to check out and check in commuter licenses. If the primary license server is down, end users will not be able to check out and check in licenses.

To set the percentage of available commuter licenses

1. Create an **LSERVOPTS** environment variable on the license manager server. This variable is created during the license manager installation on Windows, so you need to complete this step only for the other operating systems.
2. Edit the value of the **LSERVOPTS** environment variable to include **-com <percentage>**, where **<percentage>** is a numeric value between 0 and 100 indicating the percentage of licenses that are available for commuting. Specifying 0 disables commuter licenses. This switch is included by default on Windows and is set to 0.
3. Restart the computer on which the license manager is running.

To set the maximum duration for commuter licenses

The maximum length of time a user can check out a commuter license is specified by the **CommuterMaxLife** setting in the spssprod.inf file (on Windows) or the commutelicense.ini file (on Mac OS) on the *desktop* computer. On **Windows**, this file is located in the product installation directory on the desktop computer. On **Mac OS**, this file is located in the product installation directory on the desktop computer under Resources/Activation. Open the file and look for **CommuterMaxLife**. Set the value of this option to the maximum number of days for which an end user can check out a commuter license. This should be a number between 1 and 30. You can also set this value when you push the installation. See the topic [Pushing an installation to Windows computers](#) for more information.

Note: This functionality works off the date, not the time. For example, if you set the **CommuterMaxLife** option to one day, then check a license out at 9 a.m., this license does not get checked back in until midnight on the following day. So although **CommuterMaxLife** is set to one day, the license is actually held for 39 hours.

To obtain a list of checked out licenses from the command line

You can find out which users have checked out licenses.

1. Using a command prompt, change to the directory in which the license manager administrator is installed.
2. Type `lsmon <server>` (Windows) or `./lsmon <server>` (other operating systems), where `<server>` is the name or IP address of the server on which the license manager is running. You can omit the license server name if you want to view checked out licenses for only the localhost server.

To obtain a list of checked out licenses from the WlmAdmin application

You can also view checked out licenses in the WlmAdmin application:

1. In the left pane of the WlmAdmin application, click the + sign next to the license manager server.
2. Click the + sign next to Clients. Clients using the concurrent license are listed. If no clients are listed, no users are using the concurrent license.
3. Select a particular Client to view whether the client has a checked out license. Review the Detailed Information area in the right pane after selection.

Configuring license reservations

You can create a reservation file, which specifies how many licenses are reserved for specific users or groups of users. Users are identified by network IDs or computer names (*not* IP addresses). For example, you can set up a reservation file that reserves the license for a group of power users. Licenses will always be available for these users. You can also use the reservations to prevent certain users from accessing the license.

To create a new reservation file

1. From the WlmAdmin application menus, choose:
`Edit > Reservation File`
This action opens the Wlsgrmgr application.
2. From the Wlsgrmgr application menus, choose `File > New`.

To add licenses and users to the reservation file

1. From the Wlsgrmgr application menus, choose:
`Feature > Add`
2. Click Next on the first screen of the wizard.
3. Specify the feature code associated with the license that you want to reserve. Refer to [Viewing details about a license](#) for information about getting the feature codes associated with licenses. Also define a specific version, which is entered as it appears in the WlmAdmin application (for example, 160). The version is not optional. Ignore the Capacity controls, because capacity licenses are not supported.
4. Click Next.
5. Specify a name for a group of users. The name is arbitrary, but you should make it descriptive (for example, Sales).
6. Specify the number of licenses that are reserved for the group. Group members can still access all licenses, but the number of licenses you specify will no longer be available for users who are not included in the group. That is, if you have 10 licenses and you reserve five, members of the group have 10 licenses available to them, while other users have only five.
7. On the Members window, click Add to specify a user or computer name associated with the group (do not use IP addresses). If the user or machine is included in the group, that user or machine can use the reserved license. If the user or machine is excluded from the group,

that user or machine cannot access the license at all. Specify as many users or machines as needed. Note that groups must be mutually exclusive. Therefore, different groups for the same license cannot contain common users or computers.

8. Click Finish when all users have been added to the group.
9. Add other groups or licenses as needed. You can also modify or delete licenses and groups by right-clicking one and choosing Properties.

To save the reservation file

1. When finished defining the reservation file, from the menus choose:
File > Save As
2. Save the file to an easily accessible location. You will need to copy the file in the next step.
3. To enable the license server to automatically find the *lserv* file at start up, copy the file to the *winnt* subfolder of the license manager installation directory on Windows. If you accepted the default location, check the C:\Program Files (x86)\Common Files\SafeNet\Sentinel\Sentinel RMS License Manager folder. On other operating systems, copy the file directly to the license manager installation directory.
4. If you want the same reservations to apply for all redundant servers, copy the reservation file (*lserv*) to each server.
5. Restart each license manager when finished.

Starting and stopping the license manager

The method for starting the license manager depends on your operating system.

Windows

On Windows machines, the license manager is a System Service. The service is automatically started by default. However, if you need to start it manually:

1. In the Windows Control Panel, double-click Administrative Tools.
2. Double-click Services.
3. Locate Sentinel RMS License Manager in the Services list.
4. Right-click the service and choose Start or Stop.

Other operating systems

On other operating systems, the license manager is a daemon service. Complete the following steps to start it manually. You can also configure the license manager to start automatically (instructions follow).

1. Using the command prompt, browse to the directory in which you installed the license manager.
 2. **Starting.** As root, type `./lserv &` at the command prompt and press Enter.
 3. **Stopping.** As root, type `./lsrvdown <hostname>` at the command prompt, where `<hostname>` is the network name of the computer on which the license manager is running. Then press Enter.
- [Configuring the license manager to start automatically](#)

Configuring the license manager to start automatically

Windows

1. In the Windows Control Panel, double-click Administrative Tools.
2. Double-click Services.
3. Locate Sentinel RMS License Manager in the Services list.
4. Right-click the service and choose Properties.
5. Set the Startup type to Automatic.
6. Click OK.

Other operating systems

1. Add `./lserv &` to one of the operating system startup files.

Uninstalling the license manager

Windows

1. From the Windows Start menu, choose:
[Settings > Control Panel](#)
2. Double-click Add/Remove Programs.
3. Select Sentinel RMS License Manager and then click Remove.
4. Click Yes when prompted to remove the license manager.

Other operating systems

1. Using the command prompt, browse to the directory to which you installed the license manager.
2. As root, stop the license manager by typing `./lsrvdown <hostname>` at the command prompt, where `<hostname>` is the network name of the computer on which the license manager is running. Then press Enter.
3. Remove the directory where the license manager is installed.

Uninstalling the license manager administrator

1. From the Windows Start menu, choose:
[Settings > Control Panel](#)
2. Double-click Add/Remove Programs.
3. Select License Tools and then click Remove.
4. Click Yes when prompted to remove the concurrent licensing tools.

Troubleshooting Desktop Computers

If the end users' desktop computers are having trouble finding the license manager:

1. Run `lswhere` to verify that the desktop computer can find the network computer on which the license manager is running. See the topic [Running lswhere](#) for more information.
 2. Make sure the license manager service is running on the network computer.
 3. Check the appropriate `spssprod.inf` file. On Windows, this file is located in the product installation directory on the desktop computer. Open `spssprod.inf` and make sure `DAEMONHOST` is set to the correct name or IP address of the computer on which the license manager is running. If you are using redundant servers, this should define all of them. Each name is separated by a tilde (~) character. For example, if the license manager computers are `SERVER1`, `SERVER2`, and `SERVER3`, `DAEMONHOST` is set to `SERVER1~SERVER2~SERVER3`.
- [Running lswhere](#)

Running lswhere

You can run `lswhere` from an end user's desktop computer to check which computer is running the concurrent license manager.

1. Using a command prompt, change the current directory to the IBM® SPSS® Modeler installation directory.
2. Type `lswhere`.

Administrator's guide

The following instructions are for administrators at sites with the license type authorized user license for IBM® SPSS® Modeler 18.4.0. This license allows you to install IBM SPSS Modeler on multiple computers, limited to the number for which you purchased the license.

- [Before you start](#)
- [Installing the product on the local desktop computers](#)

Before you start

You will need your authorization code. The authorization code enables you and your end users to get a license for IBM® SPSS® Modeler.

You should have received separate instructions for obtaining your authorization code. If you cannot find your authorization code, contact Customer Service by visiting <https://www.ibm.com/products/spss-modeler/support>.

- [Citrix and Terminal Services](#)

Citrix and Terminal Services

You need a concurrent license to use IBM® SPSS® Modeler on Terminal Services. Contact IBM Corp. for information about transferring your license.

Installing the product on the local desktop computers

You have two options for installing the full product locally on an end user's computer. You can manually install on each computer, or you can use an application like Systems Management Server (SMS) to push the installation to the computers running Windows.

To manually install on a local desktop

1. **Make the installation media available.** Download the eImage for the product and extract the files to a shared network drive.
2. **Copy the installation instructions and prepare licensing information.** Make as many copies of the product installation instructions as you need. The installation instructions are available from the download site. Look for the instructions that correspond to your license type. After installation, the end user must enter the authorization code for your site. Fill out this information in the space provided at the beginning of the instructions before copying them.
Note: If a proxy server is preventing authorization, consider using *licenseactivator*. This allows you to enter the proxy ID and password. See the topic [Using licenseactivator](#) for more information.
3. **Distribute the installation materials to end users.** Distribute the downloaded file (or network location), the installation instructions, and the licensing information to end users who can manually install on each computer as needed.

To push to the local desktops running Windows

Because IBM® SPSS® Modeler installations are compatible with Microsoft Windows Installer (MSI), you can push an installation to the end-user desktop computers.

- [Pushing an installation to Windows computers](#)
- [Using licenseactivator](#)
- [License File](#)

Pushing an installation to Windows computers

Pushing an installation is a method for remotely distributing software to any number of end users without any user intervention. You can push the full installation of IBM® SPSS® Modeler to the end-user desktop computers running Windows. The technology that you are using for pushing the installation must support the MSI 3.0 engine or higher.

- [Uninstalling a Previous Version](#)
- [Properties for push installations](#)
- [MSI files](#)
- [Command line example](#)
- [Using SMS to push the installation](#)
- [Using Group Policy or related technology to push the installation](#)
- [Pushing an uninstallation](#)

Uninstalling a Previous Version

If you are going to push to the same directory in which a previous version of IBM® SPSS® Modeler was installed, you need to uninstall the old version. You must manually uninstall any IBM SPSS Modeler versions prior to 11.0 since push installations were not available for those versions. You can push the uninstallation as you push an installation. See the topic [Pushing an uninstallation](#) for more information.

Properties for push installations

Following are the properties that you can use for push installations. All properties are case sensitive. Values must be quoted if they contain spaces.

Table 1. Properties for push installations

Property	Description	Valid values	Default (If Applicable)
INSTALLDIR	The directory where IBM® SPSS® Modeler should be installed on the end user's desktop computer. This property is optional. If it is excluded, the default is C:\Program Files\IBM\SPSS\Modeler\18.4.0 .	A valid path such as C:\Program Files\IBM\SPSS\Modeler\18.4.0.	C:\Program Files\IBM\SPSS\Modeler\18.4.0
AUTHCODE	The authorization code. If this property is specified, the product is authorized automatically using the authorization code. If this property is <i>not</i> specified, each end user must run the License Authorization Wizard to authorize manually.	One or more valid authorization codes. Multiple authorization codes are separated by colons (for example, authcode1:authcode2).	
PROXY_USERID	The user ID for the proxy. This parameter is necessary when you specify the AUTHCODE parameter and your site is using a proxy that requires a user ID and password to connect to the Internet. This parameter works only if the Local Area Network (LAN) settings in the Internet Settings control panel reference a specific proxy server address and port.	A valid proxy user ID.	
PROXY_PASSWORD	The password for the proxy user. Refer to the discussion of PROXY_USERID for more information.	A password associated with the proxy user ID.	

MSI files

Extract the contents of your downloaded eImage to access the MSI file. The file is located under the *modeler|<architecture>* directory, where <architecture> is 32bit or 64bit.

Command line example

Following is a command line that you could use to push a product installation. Enter all of the text on one line.

```
MsiExec.exe /i "modelerclient.msi" /qn /L*v logfile.txt  
INSTALLDIR="C:\Program Files\IBM\SPSS\Modeler\18.4.0" AUTHCODE="3241a2314b23c4d5f6ea"
```

Using SMS to push the installation

The basic steps for using Systems Management Servers (SMS) to push IBM® SPSS® Modeler are:

1. Extract the contents of your downloaded eImage and copy the appropriate subdirectory under the *modeler|<architecture>* directory to a directory on a network computer.
2. Edit the *.pdf* file located in the copied directory. Using a text editor, modify the value of CommandLine by adding the appropriate properties. For a list of the available properties, refer to [Properties for push installations](#) . Make sure to specify the correct MSI file in the command line.
3. Create a package from the *.pdf* file and distribute the package to the end-user desktop machines.

Using Group Policy or related technology to push the installation

1. Extract the contents of your downloaded eImage and copy the appropriate subdirectory under the *modeler|<architecture>* directory to a directory on a network computer.
 2. Using an application like ORCA, edit the Properties table in the appropriate file under the copied folder. ORCA is part of the Windows 2003 Server SDK, which you can find at <http://www.microsoft.com/downloads> by searching for the SDK. For a list of the properties that you can add to the Properties table, refer to [Properties for push installations](#). Make sure to use the correct MSI file.
 3. Create a package using the edited file and distribute the package to the end-user desktop computers.
-

Pushing an uninstallation

Note: When you push the uninstall command, the end user loses customizations. If specific users require customizations, you can exclude those users from the distribution and ask them to install the product manually.

If you push an installation of a later version of IBM® SPSS® Modeler, you may want to uninstall first. You can do this silently by pushing the following command. Enter all of the text on one line.

```
MsiExec.exe /X{} /qn /L*v logfile.txt  
ALLUSERS=1 REMOVE="ALL"
```

The product code for a specific version is in the *setup.ini* file within each version's installed folders.

Using licenseactivator

licenseactivator allows you to authorize end-user computers without using the License Authorization Wizard. This command-line tool is located in the directory in which you installed IBM® SPSS® Modeler.

When you use *licenseactivator*, it licenses the product and writes a log file to its directory. The name of the log file is *licenseactivator_<month>_<day>_<year>.log*. If any errors occur, you can check the log file for more information. This information is also useful if you contact IBM Corp. for support.

Using licenseactivator with Authorization Codes

licenseactivator is typically used with one or more authorization codes that you received when you purchased the product. Enter all of the text on one line.

Windows:

```
licenseactivator authcode1[:authcode2:...:authcodeN] [PROXYHOST=proxy-hostname] [PROXYPORT=proxy-port-number]  
[PROXYUSER=proxy-userid] [PROXPASS=proxy-password]
```

UNIX/Linux/MacOS:

```
./licenseactivator authcode1[:authcode2:...:authcodeN] [PROXYHOST=proxy-hostname] [PROXYPORT=proxy-port-number]  
[PROXYUSER=proxy-userid] [PROXPASS=proxy-password]
```

- Multiple authorization codes are separated by colons (:).
- The proxy settings are optional, but you may need them if your computer is behind a proxy. Which proxy settings are needed depend on your specific proxy configuration. You might need all of them.

PROXYHOST

The server name or IP address of the proxy host

PROXYPORT

The port number for connecting to the Internet through the proxy

PROXYUSER

If required, the user ID for the proxy

PROXPASS

If required, the password associated with the user ID

Using licenseactivator with License Codes

In less common scenarios, IBM Corp. may have sent you a *license*.

Windows:

```
licenseactivator licensocode[:licensocode2:...:licensocodeN]
```

UNIX/Linux/MacOS:

```
./licenseactivator licensecode[:licensecode2:...:licensecodeN]
```

- Multiple license codes are separated by colons (:).
 - When using license codes, *licenseactivator* does not connect to the Internet, so you do not need to specify proxy information.
-

License File

Licensing the product creates a file called *lservrc* in the product installation directory. You can maintain a copy of this file for each end-user computer. Although the license file will work only on the computer for which it was created, the copy can be useful when there is a need to uninstall and reinstall the product. After reinstalling, you can copy the *lservrc* file back into the product installation directory. This step allows you to avoid relicensing the product.

IBM® SPSS® Modeler Scoring Adapter Installation

- [IBM SPSS Modeler scoring adapter installation](#)
-

IBM® SPSS Modeler scoring adapter installation

For some databases it is possible to enable SQL pushback of the majority of the SPSS® Modeler model nuggets. In this way, model scoring can be performed within the database, avoiding the need to extract the data before scoring. This pushback can either use the native SQL within SPSS Modeler or, where available, use additional SQL scoring adapters that are tailored for different databases.

The scoring adapters support scoring of most model nuggets in a stream, with the following exceptions:

- Association Rules (with list data), TimeSeries, Sequence, PCA, STP, and TCM are not supported.
- Association models do not support transactional format.
- Text Analytics (TA) - Field mode is supported in Db2 LUW but not the other databases. When using the TA scoring adapter in field mode, the returned row length may contain many scoring output columns, which can result in some database limits being exceeded. For example, you might see the error message TOO MANY ITEMS RETURNED IN SELECT OR INSERT LIST. Sometimes these limits can be changed in the database but, if that is not possible, the recommended approach is to use record mode to score the model.

The use of scoring adapters that enable data to be scored by the generated models within the database to avoid data transfer. SPSS Modeler enables integration with IBM and non-IBM databases, and enable models to be deployed faster and with greater efficiency.

Where scoring adapters are installed into the relevant databases the SQL generation option generates scoring adapter SQL by default, unless you specifically choose to override this. The databases for which scoring adapters are available are:

- Netezza
 - Teradata
 - Db2 LUW
 - [About scoring](#)
 - [Migrating to a new version](#)
 - [Installing IBM SPSS Modeler Server Scoring Adapter for Netezza](#)
 - [Installing IBM SPSS Modeler Server Scoring Adapter for Teradata](#)
 - [Installing IBM SPSS Modeler Server scoring adapter for Db2 LUW](#)
-

About scoring

In IBM® SPSS® Modeler, scoring data is defined as deploying a predictive model on new data with an unknown outcome. This predictive model processes incoming data and gives a predictive score about the likelihood or probability of an event. For example, when an online payment transaction takes place, a predictive model processes the input data and provides a predictive score which gives the probability of a transaction being either genuine or a fraud.

The normal process within SPSS Modeler is that when a predictive model receives incoming data it evaluates the input using historical data from a database and creates an output of a predicted score. This score gives a probability about an event for which a predictive analysis model is built.

The predictive model process using a scoring adapter differs from this in that the scoring adapter enables the evaluation of each record and for producing a score, or prediction, in the database without the need to export the data from database, run it through the model, and import it back again thereby making the whole process quicker.

Migrating to a new version

If you're upgrading to a new version of the IBM® SPSS® Modeler Server Scoring Adapter, take note of the following information.

- Adapter migration is supported from the previous version only. Migration from older versions is not supported. For example, for IBM SPSS Collaboration and Deployment Services 8.1.1 with Modeler Adapter 18.1.1, migration from IBM SPSS Collaboration and Deployment Services 7 with Modeler Adapter 17.1 is supported.
- Modeler Adapter can be installed at the same time as IBM SPSS Collaboration and Deployment Services or after IBM SPSS Collaboration and Deployment Services installation.

For complete details and instructions regarding migration, see the *Migration* section of the IBM SPSS Collaboration and Deployment Services Repository Server installation and configuration guide. The migration process for Modeler Adapter is the same as the process described in the guide. The guide is available in the IBM SPSS Collaboration and Deployment Services IBM Documentation at <https://www.ibm.com/docs/SS69YH>.

Installing IBM® SPSS® Modeler Server Scoring Adapter for Netezza

If you have a previous version of the scoring adapter for Netezza installed, first you must uninstall it as follows:

1. Locate the executable file named Uninstall IBM SPSS Modeler Server Scoring Adapter for Netezza in the Netezza scoring adapter's installation directory. It's in a folder named Uninstall IBM SPSS Modeler Server Scoring Adapter for Netezza.
2. Run the executable file either from the console or by using the graphical user interface (GUI). Follow the instructions in the uninstaller to uninstall the scoring adapter.
3. If you receive a message that some items could not be removed, go to the root directory where the adapter had been (the cfscoring directory, for example) and run the command `rm -rf` on the listed directories that were not removed. This will remove them.
4. Proceed with the following steps to install the new version of the scoring adapter.

Depending on the configuration of your database you can install either from the console or by using a graphical user interface (GUI); however, the first step is the same for both methods:

- Run the *install.bin* install script. Ensure that *install.bin* can be executed by *nz* user and run it as that user.

Console Installation

1. Introduction details are displayed. Press Enter to continue.
2. Licensing information is displayed. Read the license, type *Y* to accept it, and press Enter to continue.
3. You are prompted to type the installation location. The default installation location is shown; however, if your installation is different, type the revised location and press Enter.
Note: The installation must be under the /nz/export/ path.
4. You are prompted to enter the database name, database user name, and database password.
Note: The database user must have database access permissions to initialize the database and register the udf modules.
5. A pre-installation summary is displayed to confirm your entries so far. Press Enter to continue.
6. A message is displayed to say the installation routine is ready to run. Press Enter to continue.
7. A progress bar is displayed whilst the installation routine runs. When the installation is complete, press Enter to exit from the installer.

GUI Installation

1. Introduction details are displayed. Click Next to continue.
2. Licensing information is displayed. Read the license, select the option to accept it, and click Next to continue.
3. You are prompted to select the installation location. The default installation location is shown; however, if your installation is different, click Choose to browse for the revised location. When the correct location is shown, click Next.
4. You are prompted to enter the database name, database user name, and database password.
Note: The database user must have database access permissions to initialize the database and register the UDF modules.
5. A pre-installation summary is displayed to confirm your entries so far. Click Install to continue.
6. A progress bar is displayed whilst the installation routine runs. When the installation is complete, click Done to exit from the installer.

When you have completed these steps the scoring adapter is ready to receive work.

Note: The Netezza UDF has a 64 field limit that can be processed by the scoring adapter. If that field limit is exceeded, the validation error message: SQL Validation Error: HY000[46] ERROR: Cannot pass more than 64 arguments to a function is displayed and model scoring continues without using the scoring adapter.

Installing IBM SPSS Modeler Server Scoring Adapter for Teradata

If you have a previous version of the scoring adapter for Teradata installed, first you must uninstall it as follows:

1. Locate the executable file named Uninstall IBM SPSS Modeler Server Scoring Adapter for Teradata in the Teradata scoring adapter's installation directory. It's in a folder named Uninstall IBM SPSS Modeler Server Scoring Adapter for Teradata.
2. Run the executable file either from the console or by using the graphical user interface (GUI). Follow the instructions in the uninstaller to uninstall the scoring adapter.
3. If you receive a message that some items could not be removed, go to the root directory where the adapter had been (the cfscoring directory, for example) and run the command `rm -rf` on the listed directories that were not removed. This will remove them.
4. Proceed with the following steps to install the new version of the scoring adapter.

Depending on the configuration of your database you can install either from the console or by using a graphical user interface (GUI); however, the first step is the same for both methods:

- Log in as either *root* or *DBA user* and run the *install.bin* install script. You must have access permissions for the installation folder to do this. The installing user must also have **CREATE FUNCTION** permissions.

Console Installation

1. Introduction details are displayed. Press Enter to continue.
2. Licensing information is displayed. Read the license, type **Y** to accept it, and press Enter to continue.
3. You are prompted to type the installation location. The default installation location is shown; however, if your installation is different, type the revised location and press Enter.
4. Enter the database TDPID. Press Enter to continue.
5. Enter the user name. Press Enter to continue.
6. Enter the password. Press Enter to continue.
7. A pre-installation summary is displayed to confirm your entries so far. Press Enter to continue.
8. A message is displayed to say the installation routine is ready to run. Press Enter to continue.
9. A progress bar is displayed whilst the installation routine runs. When the installation is complete, press Enter to exit from the installer.
10. If the *Components* table exists in your database a confirmation message is displayed. Either enter **Y** to continue creating tables and functions in your database, or enter **N** to skip this step. *Note:* If you skip this step you must manually create tables and functions later using initdb.sh, which is stored in the <installation path>\setup folder.

GUI Installation

1. Introduction details are displayed. Click Next to continue.
2. Licensing information is displayed. Read the license, select the option to accept it, and click Next to continue.
3. You are prompted to select the installation location. The default installation location is shown; however, if your installation is different, click Choose to browse for the revised location. When the correct location is shown, click Next.
4. Enter the database TDPID, name, and password and click Next to continue.
5. A pre-installation summary is displayed to confirm your entries so far. Click Install to continue.
6. A progress bar is displayed whilst the installation routine runs. When the installation is complete, click Done to exit from the installer.
7. If the *Components* table exists in your database a confirmation message is displayed. Either click Yes to continue creating tables and functions in your database, or click No to skip this step. *Note:* If you skip this step you must manually create tables and functions later using initdb.sh, which is stored in the <installation path>\setup folder.

When you have completed these steps the scoring adapter is ready to receive work.

Note: The UDFs and COMPONENTS table are installed into the default database of the user who installs the scoring adapter.

Sharing the scoring adapter

To share the scoring adapter for use by other Teradata users:

1. Grant the following privileges to the user:
 - SELECT and EXECUTE FUNCTION on the database where the scoring adapter is installed.
 - INSERT on the COMPONENTS table on the database where the scoring adapter is installed.
2. When a database connection is made to Teradata with the scoring adapter installed, open the Database Preset dialog box, enable Use Server Scoring Adapter Schema and select the schema from the Server Scoring Adapter Schema drop down list.

Note: The Database Preset dialog box varies for different databases and is not supported by scripting; therefore, this step can only be done in SPSS® Modeler Client.

Preventing SQL errors for date or time entries

If the ODBC driver for the Date or Time format is set to Integer, and your input table contains fields that are coded as Date, Time, or Timestamp; Teradata displays a SQL error message and is unable to process those fields.

To prevent this error from happening follow these steps:

1. Open the ODBC Data Source Administrator.
2. Open the DSN that uses the Teradata driver.
3. Click on Options >> to open the Teradata ODBC Driver Options dialog box.
4. On the top right side of the dialog box, set the Date Time Format to AAA.
5. Save your changes.
6. In IBM® SPSS Modeler Server, remove the connection and then reconnect to the DSN that uses the Teradata driver.

Note: On Unix/Linux, the option is called: DateTimeFormat.

Teradata drivers and null datetime_now values

When using the DataDirect Teradata driver, the `SQL_COLUMN_TYPE` may return a null value. The alternative is to use the native Teradata driver.

Installing IBM SPSS Modeler Server scoring adapter for Db2 LUW

Note: The Db2 LUW scoring adapter is only available on Db2 running on LINUX or AIX.

Note: The IBM® SPSS® Modeler Server Scoring Adapter may conflict with the Db2 LUW ANALYZE_TABLE embedded process for SAS because they share the same Db2 built in support.

If you have a previous version of the scoring adapter for Db2 LUW installed, first you must uninstall it as follows:

1. Locate the executable file named Uninstall IBM SPSS Modeler Server Scoring Adapter for DB2 in the Db2 scoring adapter's installation directory. It's in a folder named Uninstall IBM SPSS Modeler Server Scoring Adapter for DB2.
2. Run the executable file either from the console or by using the graphical user interface (GUI). Follow the instructions in the uninstaller to uninstall the scoring adapter.
3. If you receive a message that some items could not be removed, go to the root directory where the adapter had been (the cfscoring directory, for example) and run the command `rm -rf` on the listed directories that were not removed. This will remove them.
4. Proceed with the following steps to install the new version of the scoring adapter.

Before installation, you need to shut down the Db2 LUW ANALYZE_TABLE embedded process using the `db2ida_epspss.sh` script provided in the IBM SPSS Modeler Server scoring adapter installation folder. To do this, use the Db2 command: `db2ida_epspss.sh stop`.

After installation, the Db2 LUW ANALYZE_TABLE embedded process should start automatically using `db2start`. However, if you install the adapter while a Db2 instance is active, you can start the Db2 LUW ANALYZE_TABLE embedded process manually by using the command: `db2ida_epspss.sh start`.

Note: The Db2 LUW ANALYZE_TABLE doesn't support the WITH table-expression clause if it contains UNION ALL. This may cause an error if you attempt to use this expression in a IBM SPSS Modeler node which generates UNION or UNION ALL SQL, such as the Append node.

Depending on the configuration of your database you can install either from the console or by using a graphical user interface (GUI); however, the first step is the same for both methods:

- Run the `install.bin` install script. Ensure that `install.bin` can be executed by db2 user and run it as that user.

Console Installation

1. Introduction details are displayed. Press Enter to continue.
2. Licensing information is displayed. Read the license, type 1 to accept it and press Enter.
3. You are prompted to type the installation location. The default installation location is shown; however, if your installation is different, type the revised location and press Enter.
4. You are prompted to enter the database name, database user name, and database password.
5. A pre-installation summary is displayed to confirm your entries so far. Press Enter to continue.
6. A message is displayed to say the installation routine is ready to run. Press Enter to continue.
7. A progress bar is displayed whilst the installation routine runs. When the installation is complete, press Enter to exit from the installer.

GUI Installation

1. Introduction details are displayed. Click Next to continue.
2. Licensing information is displayed. Read the license, select the option to accept it, and click Next to continue.
3. You are prompted to select the installation location. The default installation location is shown; however, if your installation of is different, click Choose to browse for the revised location. When the correct location is shown, click Next.
4. You are prompted to enter the database name, database user name, and database password.
5. A pre-installation summary is displayed to confirm your entries so far. Click Install to continue.
6. A progress bar is displayed whilst the installation routine runs. When the installation is complete, click Done to exit from the installer.

When you have completed these steps the scoring adapter is ready to receive work.

Note: If you have problems scoring large Text Mining models via the Database Scoring Adapters for Db2 LUW, you may need to modify the database table column size parameter. If you encounter errors related to failing to insert the model into the components table, use a Db2 command such as the following to increase the column size parameter as appropriate:

```
ALTER TABLE COMPONENTS ALTER COLUMN MODELDS2 SET DATA TYPE BLOB (48M) ;
```

Installation instructions

The following instructions are for installing IBM® SPSS® Modeler Server version 18.4.0.

IBM SPSS Modeler Server can be installed and configured to run in distributed analysis mode together with one or more client installations. This provides superior performance on large datasets, since memory-intensive operations can be run on the server without downloading data to the client computer. At least one IBM SPSS Modeler Client installation must be present to run an analysis.

Whenever you install a new version, be sure to distribute IBM SPSS Modeler Server product's host name and port number to the end users.

- [System requirements](#)
- [Installing](#)
- [After You Install IBM SPSS Modeler Server](#)
- [Uninstalling](#)

System requirements

To view system requirements, go to <https://www.ibm.com/software/reports/compatibility/clarity/softwareReqsForProduct.html>.

- [Additional requirements](#)

Additional requirements

Client software

The client software must be at the same release level as the IBM® SPSS® Modeler Server software.

You must ensure that kernel limits on the system are sufficient for the operation of IBM SPSS Modeler Server. The data, memory, file, and processes ulimits are particularly important and should be set to unlimited within the IBM SPSS Modeler Server environment. To do this:

1. Add the following commands to modelersrv.sh:

```
ulimit -d unlimited  
ulimit -m unlimited  
ulimit -f unlimited  
ulimit -u unlimited
```

In addition, set the stack limit to the maximum allowed by your system (`ulimit -s XXXX`), for example:

```
ulimit -s 64000
```

2. Restart IBM SPSS Modeler Server.

You also need the `gzip` file compression utility and `GNU cpio` installed and on the PATH in order for the installer to be able to decompress the installation files. In addition, on the machine running SPSS Modeler Server, you should set the locale to EN_US.UTF-8.

XL C++ runtime and XL FORTRAN runtime

If installing on PowerLinux, for SPSS Modeler Server to start correctly on P8LE servers (Ubuntu and RedHat), you must install the XL C++ runtime and the XL FORTRAN runtime (version 16.1.1 or later). If these libraries aren't installed, SPSS Modeler Server won't start.

To install the **XL C/C++ runtime** as root, issue the following commands:

```
1. wget
   http://public.dhe.ibm.com/software/server/POWER/Linux/rte/xlcpp/le/rhel7/repo/repodata/repomd.xml.key
2. rpm --import repomd.xml.key
3. wget
   http://public.dhe.ibm.com/software/server/POWER/Linux/rte/xlcpp/le/rhel7/ibm-xlc-compiler-runtime.repo
4. cp ibm-xlc-compiler-runtime.repo /etc/yum.repos.d/
5. yum install libxlc libxlsmp
```

To install the **XL FORTRAN runtime** as root, issue the following commands:

```
1. wget
   http://public.dhe.ibm.com/software/server/POWER/Linux/rte/xlf/le/rhel7/repo/repodata/repomd.xml.key
2. rpm --import repomd.xml.key
3. wget
   http://public.dhe.ibm.com/software/server/POWER/Linux/rte/xlf/le/rhel7/ibm-xlf-compiler-runtime.repo
4. cp ibm-xlf-compiler-runtime.repo /etc/yum.repos.d/
5. yum install libxlf
```

Installing

You can install IBM® SPSS® Modeler Server as *root* or as a non-root user. If your site restricts the use of the *root* password, use an authentication method that supports running as non-root (see the IBM SPSS Modeler Server and Performance Guide). Then install the product as the user who will run the daemon. You should perform all actions as non-root, or perform all actions as root. Note that you need the root password to start and stop the server.

Note: The installation will fail if you attempt to install SPSS Modeler Server on Linux as a user who doesn't have execute permissions for files in /tmp. To avoid this, you must either have execute permissions to files within /tmp for the InstallAnywhere SPSS Modeler installs to succeed or, if this is not present on your environment, you can set and export *IATEMPDIR* to a location where you do have permissions in order to run the install.

Important: The file system on which you install IBM SPSS Modeler Server must be mounted with the **suid** option. The product will not work correctly if the file system is mounted with the **nosuid** option.

1. **From your downloaded installation media, extract the installation files.** The downloaded media file is a compressed archive. Extract the files in the archive.
2. **From your downloaded installation media, run the installation file.** The downloaded media contains a *.bin* file; run this file.
3. **Check hard drive space.** In addition to the permanent hard drive space specified in <http://www.ibm.com/software/analytics/spss/products/modeler/requirements.html>, you need temporary disk space for the installer files. The installer files are extracted to your system's temporary folder. If there is not enough space in the temporary folder, the installer files are extracted to your home folder. If neither location has enough space, the installer cannot continue. In this case, you can temporarily set the *IATEMPDIR* environment variable to a location with adequate space. This location should have at least 2.5 gigabyte (GB) of free space.
4. **Check the destination directory.** By default, IBM SPSS Modeler Server is installed to */usr/IBM/SPSS/ModelerServer/<version>* . If desired, you can change this path in the graphical installation wizard or the command line installation. If you are going to run the silent installer, you can set the value for *USER_INSTALL_DIR* in *installer.properties*. Regardless, you need read and write permissions to the installation directory, so log on with an account that has sufficient permissions. *Note:* If you are upgrading by adding a new version of the product, install the new version in a separate directory.
5. **Change execute permissions of installer.** Be sure that the installer is executable by the user who will run the installer.
6. **Run the installer.** You can run the installer in a graphical user interface, from the command line, or silently. Instructions for each method appear below.
 - [Graphical installation wizard](#)
 - [Command line installation](#)
 - [Silent installation](#)

Graphical installation wizard

The graphical installation wizard displays a graphical interface that will ask you about installation parameters. You will need an X Window System.

1. At the UNIX prompt, change to the directory where the installer file was copied or extracted:
2. Run the installer by executing the following command:
 `./<installer_name>`
3. After the installation wizard is launched, follow the instructions that appear on the screen.

Command line installation

The command line installation uses command prompts to specify installation parameters.

1. At the UNIX prompt, change to the directory where the installer file was copied or extracted:
2. Run the installer by executing the following command:
 `./<installer_name> -i console`

Where `<installer_name>` is the installer .bin file.

3. Follow the instructions that appear on the screen.

Silent installation

Silent mode enables the installation without any user interaction. Installation parameters are specified as a properties file.

To complete a silent installation on Linux or UNIX systems:

1. In the same location where you copied the installer files, create an `installer.properties` file.
2. In a text editor, set the `installer.properties` values. The following text shows an example of an `installer.properties` file:

```
=====
# Thu Jan 29 11:35:37 GMT 2015
# Replay feature output
#
# -----
# This file was built by the Replay feature of InstallAnywhere.
# It contains variables that were set by Panels, Consoles or Custom Code.

#Indicate whether the license agreement been accepted
#-----
LICENSE_ACCEPTED=TRUE

#Server Mode
#-----
SERVEMODE_SELECT_OPTION=\"1\"

#Choose Install Folder
#-----
USER_INSTALL_DIR=/usr/IBM/SPSS/ModelerServer/17.0

#Install
=====
```

3. The value for `SERVEMODE_SELECT_OPTION` depends on the type of installation you have. You can choose from the following values:
 - 0 - Non-Production Mode. If you purchased a separate non-production installation, enter this option. This installation cannot be employed for production use.
 - 1 - Production Mode. A production installation is a standard installation of SPSS® Modeler Server. It is appropriate for production use.
4. Ensure that the value for `USER_INSTALL_DIR` matches your installation directory location. The directory path cannot contain spaces.
5. Save the file.
6. Run the installer by using the following command:

```
./<installer_name> -i silent -f installer.properties
```

Where `<installer_name>` is the installer .bin file.

After You Install IBM SPSS Modeler Server

This section describes some required and optional steps that you can perform after installation. It does not describe all possible configuration options. You can find information about all the configuration options in the *IBM® SPSS® Modeler Server and Performance Guide*.

Note: Installation logs are placed into the uninstall folder by default. For example:

<Installation_folder_path>/Uninstall_IBM_SPSS_MODELER_SERVER/Logs.

- [Installing IBM SPSS Modeler Batch](#)
 - [Configuring IBM SPSS Modeler to Work with IBM SPSS Statistics](#)
 - [Enabling IBM SPSS Statistics Programmability](#)
 - [Starting the Process](#)
 - [Checking the Server Status](#)
 - [Connecting End Users](#)
 - [IBM SPSS Data Access Pack Technology](#)
-

Installing IBM SPSS Modeler Batch

IBM® SPSS® Modeler Batch provides the complete analytical capabilities of the standard IBM SPSS Modeler Client but without access to the regular user interface. Batch mode allows you to perform long-running or repetitive tasks without your intervention and without the presence of the user interface on the screen. It must be run in distributed mode along with IBM SPSS Modeler Server (local mode is not supported).

For more information, see the *IBM SPSS Modeler Batch User's Guide*.

Installing IBM SPSS Modeler Batch on Linux

1. Change directories to the modelbat directory.
2. Change to the relevant platform directory.
3. Run the .bin install script. Make sure the .bin can be executed by **root**. For example:

```
./modelerserverlinux64.bin -i console
```

or:

```
./modelerbatchlinux.bin -i console
```

4. Introduction details are displayed. Press Enter to continue.
5. Licensing information is displayed. Read the license, type 1 to accept it, and press Enter to continue.
6. You are prompted to type the installation location. To use the default directory (for example: /usr/IBM/SPSS/ModelerServer/<nn> or: /usr/IBM/SPSS/ModelerBatch/<nn>, where <nn> is the version number), press Enter. If you specify a directory other than the default, make sure that the path name does not contain extended ASCII characters, the space character, or the ampersand (&) character.
7. You are prompted to confirm the installation location. When it is correct, type y and press Enter.
8. A pre-installation summary is displayed to confirm your entries so far. Press Enter to continue.
9. A message is displayed to say the installation routine is ready to run. Press Enter to continue.
10. A progress bar is displayed whilst the installation routine runs. When the installation is complete, press Enter to exit from the installer.

Configuring IBM SPSS Modeler to Work with IBM SPSS Statistics

To enable IBM® SPSS® Modeler to use the Statistics Transform, Statistics Model, and Statistics Output nodes, you must have a copy of IBM SPSS Statistics installed and licensed on the computer where the stream is run.

If running IBM SPSS Modeler in local (standalone) mode, the licensed copy of IBM SPSS Statistics must be on the local computer.

When you have finished installing this copy of SPSS Modeler Client, you will also need to configure it to work with IBM SPSS Statistics. From the main client menu, choose:

Tools > Options > Helper Applications

and on the IBM SPSS Statistics tab, specify the location of the local IBM SPSS Statistics installation you want to use. For more information, see the *Source, Process and Output Nodes* guide or the online help for Helper Applications.

In addition, if running in distributed mode against a remote IBM SPSS Modeler Server, you also need to run a utility at the IBM SPSS Modeler Server host to create the **statistics.ini** file, which indicates to IBM SPSS Statistics the installation path for IBM SPSS Modeler Server. To do this, from the command prompt, change to the IBM SPSS Modeler Server **bin** directory and, for Windows, run:

```
statisticsutility -location=<IBM SPSS Statistics_installation_path>/bin
```

Alternatively, for UNIX, run:

```
./statisticsutility -location=<IBM SPSS Statistics_installation_path>/bin
```

If you do not have a licensed copy of IBM SPSS Statistics on your local machine, you can still run the Statistics File node against a IBM SPSS Statistics server, but attempts to run other IBM SPSS Statistics nodes will display an error message.

Enabling IBM SPSS Statistics Programmability

If you have IBM® SPSS® Statistics installed and you want to be able to call its Python or R plugins through the IBM SPSS Statistics nodes in IBM SPSS Modeler Server, you must take the following steps on the UNIX server to enable the plugins.

1. Log in as the superuser.
2. Export environment variables as follows:
 - Linux. `export LD_LIBRARY_PATH=[plugin_install_directory]/lib:$LD_LIBRARY_PATH`

Starting the Process

IBM® SPSS® Modeler Server runs as a daemon process and has root privileges by default. IBM SPSS Modeler Server can be configured to run without root privileges. Refer to the *IBM SPSS Modeler Server and Performance Guide* for more information. You need to choose an authentication method that does not require that the daemon runs as root.

Start the application by running a startup script, `modelersrv.sh`, which is included in the installation directory. The startup script configures the environment for and executes the software.

1. Log in as `root`. Alternatively, you can log in as `non-root` if the non-root user is also the user who installed IBM SPSS Modeler Server.
2. Change to the IBM SPSS Modeler Server installation directory. The startup script must be run from this location.
3. Run the startup script. For example, at the UNIX prompt type:

```
./modelersrv.sh start
```

Checking the Server Status

1. At the UNIX prompt, type:
`/modelersrv.sh list`
2. Look at the output, which is similar to what the UNIX `ps` command produces. If the server is running, you will see it as the first process in the list.

IBM® SPSS® Modeler Server is now ready to accept connections from end users when they have been authorized. See the topic [Connecting End Users](#) for more information.

Connecting End Users

End users connect to IBM® SPSS® Modeler Server by logging in from the client software. See the *IBM SPSS Modeler Server and Performance Guide* for a description of how the software works and what you need to do to administer it. You must give end users the information that they need to connect, including the IP address or host name of the server machine.

IBM SPSS Data Access Pack Technology

To read or write to a database, you must have an ODBC data source installed and configured for the relevant database, with read or write permissions as needed. The IBM SPSS Data Access Pack includes a set of ODBC drivers that can be used for this purpose, and these drivers are available from the download site. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

Supported ODBC drivers

For the latest information on which databases and ODBC drivers are supported and tested for use with IBM® SPSS® Modeler, see the product compatibility matrices on the corporate Support site (<http://www.ibm.com/support>).

Where to install drivers

Note: ODBC drivers must be installed and configured on each computer where processing may occur.

- If you are running IBM SPSS Modeler in local (standalone) mode, the drivers must be installed on the local computer.
- If you are running IBM SPSS Modeler in distributed mode against a remote IBM SPSS Modeler Server, the ODBC drivers need to be installed on the computer where IBM SPSS Modeler Server is installed. For IBM SPSS Modeler Server on UNIX systems, see also "Configuring ODBC drivers on UNIX systems" later in this section.
- If you need to access the same data sources from both IBM SPSS Modeler and IBM SPSS Modeler Server, the ODBC drivers must be installed on both computers.
- If you are running IBM SPSS Modeler over Terminal Services, the ODBC drivers need to be installed on the Terminal Services server on which you have IBM SPSS Modeler installed.

Configuring ODBC drivers on UNIX systems

By default, the DataDirect Driver Manager is not configured for IBM SPSS Modeler Server on UNIX systems. To configure UNIX to load the DataDirect Driver Manager, enter the following commands:

```
cd <modeler_server_install_directory>/bin  
rm -f libspssodbc.so
```

Then run this command if you want to use the UTF8 driver wrapper:

```
ln -s libspssodbc_datadirect.so libspssodbc.so
```

Or run this command instead if you want to use the UTF16 driver wrapper:

```
ln -s libspssodbc_datadirect_utf16.so libspssodbc.so
```

Doing so removes the default link and creates a link to the DataDirect Driver Manager.

Note: The UTF16 driver wrapper is required to use SAP HANA or IBM Db2 CLI drivers for some databases. DashDB requires the IBM Db2 CLI driver.

To configure SPSS Modeler Server:

1. Configure the SPSS Modeler Server start up script modelersrv.sh to source the IBM SPSS Data Access Pack odbc.sh environment file by adding the following line to modelersrv.sh:

```
. /<pathtoSDAPinstall>/odbc.sh
```

Where <pathtoSDAPinstall> is the full path to your IBM SPSS Data Access Pack installation.

2. Restart SPSS Modeler Server.

In addition, for SAP HANA and IBM Db2 only, add the following parameter definition to the DSN in your odbc.ini file to avoid buffer overflows during connection:

```
DriverUnicodeType=1
```

Note: The libspssodbc_datadirect_utf16.so wrapper is also compatible with the other SPSS Modeler Server supported ODBC drivers.

- [Configuring IBM SPSS Modeler Server for Data Access](#)

Configuring IBM SPSS Modeler Server for Data Access

If you want to use the IBM® SPSS® Data Access Pack with IBM SPSS Modeler Server, you will need to configure the startup scripts. This is a critical step because data access will not work otherwise. For instructions, refer to the *IBM SPSS Modeler Server and Performance Guide*.

Uninstalling

Uninstall IBM® SPSS® Modeler Server by removing the program files and, if you have configured the system for automatic startup, disabling automatic startup.

To Remove the Program Files

1. Stop the server process. Instructions for stopping the server process are in the *IBM SPSS Modeler Server and Performance Guide*.

2. Remove the installation directory.

Silent removal of an installation

Silent mode enables software to be uninstalled without any user interaction. To silently remove an installation in IBM SPSS Modeler Server:

1. Run the uninstaller by executing the following command:

```
./<installer_path>/Uninstall_IBM_SPSS_MODELER_SERVER/Uninstall_IBM_SPSS_MODELER_SERVER -i silent
```

Where <installer_path> is the path name to the IBM SPSS Modeler Server installation directory.

To Disable Automatic Startup

The IBM SPSS Modeler Server includes a script that you can use to configure your system to start the server daemon automatically when the computer is rebooted (the instructions appear in the *IBM SPSS Modeler Server and Performance Guide*).

Installation instructions

The following instructions are for installing IBM® SPSS® Modeler Server version 18.4.0.

IBM SPSS Modeler Server can be installed and configured to run in distributed analysis mode together with one or more client installations. This provides superior performance on large datasets, since memory-intensive operations can be run on the server without downloading data to the client computer. At least one IBM SPSS Modeler Client installation must be present to run an analysis.

Whenever you install a new version, be sure to distribute IBM SPSS Modeler Server product's host name and port number to the end users.

- [System requirements](#)
 - [Installing](#)
 - [After You Install IBM SPSS Modeler Server](#)
 - [Uninstalling](#)
-

System requirements

To view system requirements, go to <https://www.ibm.com/software/reports/compatibility/clarity/softwareReqsForProduct.html>.

Installing

The Setup program installs the following components:

- A Windows service that manages end-user requests.
- Software that handles the data mining process .

Note: IBM® SPSS® Modeler Server must be installed on a hard drive on the computer on which the Setup program is running.

Windows Server

1. Double-click the downloaded file and extract the installation files.
2. Using Windows Explorer, browse to the location where the installation files were extracted.
3. Click on setup.exe.
Note: You must run setup.exe as administrator:

4. On the AutoPlay menu, choose Install IBM SPSS Modeler Server and then follow the instructions that appear on the screen.

- [Destination](#)
 - [Silent installation](#)
 - [IP Address and Port Number](#)
-

Destination

You can install in a different destination folder, but you must install on the computer from which the setup is being run (you cannot install to a network location).

If you are installing on the same computer with other Server products, install in a *separate* directory. Do not install multiple Server products in the same directory.

Silent installation

Silent mode enables an installation to run on its own without any interaction; installing silently can free system administrators from the task of monitoring each installation and providing input to prompts and dialog boxes. This method is especially useful when you are installing SPSS® Modeler Server on a number of different computers that have identical hardware.

Note: You must have administrator privileges to be able to run silent installations.

Windows - silent installation

You can complete a silent installation on Windows systems by using Microsoft Installer (MSI). Use msieexec.exe to install the MSI package.

The following options can be used:

Table 1. Silent installation options

Option	Description
/i	Specifies that the program is to install the product.
/l*v	Specifies verbose logging. For example, this form of log can be useful if you need to troubleshoot an installation.
/qn	Runs the installation without running the external user interface sequence.
/s	Specifies silent mode.
/v	Specifies that the Setup Program passes the parameter string to the call it makes to the MSI executable file (msieexec.exe). The following syntax requirements apply if you use this option: <ul style="list-style-type: none">• You must place a backslash (\) in front of any quotation marks (" ") that are within existing quotation marks.• Do not include a space between the /v option and its arguments.• Multiple parameters that are entered with the /v option must be separated with a space.• To create a log file, specify the directory and file name at the end of the command. The directory must exist before you start the silent installation.
/x	Specifies that the program is to uninstall the product.

The following text shows an example of the MSI command:

```
c:\>msieexec.exe /i ModelerServer64.msi /qn /l*v  
c:\temp\Modeler_Silent_Install.log  
INSTALLDIR="C:\Program Files\IBM\SPSS\ModelerServer\19"  
SERVERMOD=1
```

Where the value for SERVERMOD depends on the type of installation you have. You can choose from the following values:

- 0 - Non-Production Mode. If you purchased a separate non-production installation, enter this option. This installation cannot be employed for production use.
- 1 - Production Mode. A production installation is a standard installation of SPSS Modeler Server. It is appropriate for production use.

Windows - silent uninstalling

The following text shows an example of the MSI command to silently uninstall the software:

```
c:\>msieexec.exe /x ModelerServer64.msi /qn /norestart
```

IP Address and Port Number

The Setup program will supply a default IP address and port number for the server computer to use. If necessary, the port number can be updated in the configuration file (*options.cfg*) or with the Administration Console included in IBM® SPSS® Deployment Manager.

Invalid digital signature on installation

IBM® SPSS® Modeler products use IBM-issued certification for digital signing. In certain circumstances you may see the following error on trying to install SPSS Modeler products:

```
Error 1330. A file that is required cannot be installed because the cabinet file filename has an invalid digital signature...
```

All Windows users

You see this message if you try to install SPSS Modeler products on a machine that has no Internet connection and does not have the correct certificate installed. Use the following procedure to correct this problem.

1. Click OK to acknowledge the message.
 2. Click Cancel to exit from the installer.
 3. If the machine on which you want to install has no Internet connection, perform the next step on an Internet-connected machine and copy the .cer file to the machine where you want to install.
 4. Go to <https://support.symantec.com>, search for VeriSign Class 3 Primary Certification Authority - G5 root certificate, and download it. Save it as a .cer file.
 5. Double-click the .cer file.
 6. On the General tab, click Install Certificate.
 7. Follow the instructions in the Certificate Import Wizard, using the default options and clicking Finish at the end.
 8. Retry the installation.
-

After You Install IBM SPSS Modeler Server

This section describes some required and optional steps that you can perform after installation. It does not describe all possible configuration options. You can find information about all the configuration options in the *IBM® SPSS® Modeler Server and Performance Guide*.

Note: Installation logs are placed into the uninstall folder by default. For example:

`<Installation_folder_path>/Uninstall_IBM_SPSS_MODELER_SERVER/Logs`.

- [Installing IBM SPSS Modeler Batch](#)
 - [Configuring IBM SPSS Modeler to Work with IBM SPSS Statistics](#)
 - [Checking the Server Status](#)
 - [Connecting End Users](#)
 - [IBM SPSS Data Access Pack Technology](#)
-

Installing IBM SPSS Modeler Batch

IBM® SPSS® Modeler Batch provides the complete analytical capabilities of the standard IBM SPSS Modeler Client but without access to the regular user interface. Batch mode allows you to perform long-running or repetitive tasks without your intervention and without the presence of the user interface on the screen. It must be run in distributed mode along with IBM SPSS Modeler Server (local mode is not supported).

Follow the steps for Windows Server in [Installing](#), except, on the AutoPlay menu, choose Install IBM SPSS Modeler Batch and then follow the instructions that appear on the screen.

For more information, see the *IBM SPSS Modeler Batch User's Guide*.

Configuring IBM SPSS Modeler to Work with IBM SPSS Statistics

To enable IBM® SPSS® Modeler to use the Statistics Transform, Statistics Model, and Statistics Output nodes, you must have a copy of IBM SPSS Statistics installed and licensed on the computer where the stream is run.

If running IBM SPSS Modeler in local (standalone) mode, the licensed copy of IBM SPSS Statistics must be on the local computer.

When you have finished installing this copy of SPSS Modeler Client, you will also need to configure it to work with IBM SPSS Statistics. From the main client menu, choose:

Tools > Options > Helper Applications

and on the IBM SPSS Statistics tab, specify the location of the local IBM SPSS Statistics installation you want to use. For more information, see the *Source, Process and Output Nodes* guide or the online help for Helper Applications.

In addition, if running in distributed mode against a remote IBM SPSS Modeler Server, you also need to run a utility at the IBM SPSS Modeler Server host to create the **statistics.ini** file, which indicates to IBM SPSS Statistics the installation path for IBM SPSS Modeler Server. To do this, from the command prompt, change to the IBM SPSS Modeler Server *bin* directory and, for Windows, run:

```
statisticsutility -location=<IBM SPSS Statistics installation path>/bin
```

Alternatively, for UNIX, run:

```
./statisticsutility -location=<IBM SPSS Statistics installation path>/bin
```

If you do not have a licensed copy of IBM SPSS Statistics on your local machine, you can still run the Statistics File node against a IBM SPSS Statistics server, but attempts to run other IBM SPSS Statistics nodes will display an error message.

Checking the Server Status

1. On the computer where you installed IBM® SPSS® Modeler Server, select Services from Administrative Tools on the Control Panel.
2. Locate IBM SPSS Modeler Server in the list. If the service is not started, double-click its name and start it on the dialog box that appears. Note that if the service startup is configured to be Automatic, the service will start automatically whenever the computer is restarted.
3. Click OK to close the dialog box.

IBM SPSS Modeler Server is now ready to accept connections from end users when they have been authorized. See the topic [Connecting End Users](#) for more information.

Connecting End Users

End users connect to IBM® SPSS® Modeler Server by logging in from the client software. See the *IBM SPSS Modeler Server and Performance Guide* for a description of how the software works and what you need to do to administer it. You must give end users the information that they need to connect, including the IP address or host name of the server machine. You also need to enable local logon for end users by adding them to the local logon policy. From the Windows Control Panel, choose Administrative Tools, then Local Security Policy, then Local Policies, then User Rights Assignment, then double-click Log On Locally and add users or groups.

IBM SPSS Data Access Pack Technology

To read or write to a database, you must have an ODBC data source installed and configured for the relevant database, with read or write permissions as needed. The IBM SPSS Data Access Pack includes a set of ODBC drivers that can be used for this purpose, and these drivers are available from the download site. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

Supported ODBC drivers

For the latest information on which databases and ODBC drivers are supported and tested for use with IBM® SPSS® Modeler, see the product compatibility matrices on the corporate Support site (<http://www.ibm.com/support>).

Where to install drivers

Note: ODBC drivers must be installed and configured on each computer where processing may occur.

- If you are running IBM SPSS Modeler in local (standalone) mode, the drivers must be installed on the local computer.
- If you are running IBM SPSS Modeler in distributed mode against a remote IBM SPSS Modeler Server, the ODBC drivers need to be installed on the computer where IBM SPSS Modeler Server is installed.
- If you need to access the same data sources from both IBM SPSS Modeler and IBM SPSS Modeler Server, the ODBC drivers must be installed on both computers.
- If you are running IBM SPSS Modeler over Terminal Services, the ODBC drivers need to be installed on the Terminal Services server on which you have IBM SPSS Modeler installed.

Uninstalling

1. From the Windows Start menu choose:

Settings > Control Panel

2. From the Control Panel, choose Add/Remove Programs.
3. Click the Change or Remove Programs button on the left, choose IBM® SPSS® Modeler Server from the list, and click Change/Remove.

Note: If you have more than one version of IBM SPSS Modeler Server installed on the computer, be sure to choose the version that you want to remove.

A message will be displayed when uninstallation is complete. This may take several minutes.

IBM SPSS Modeler Premium component overview

IBM® SPSS® Modeler Premium includes IBM SPSS Modeler Text Analytics.

IBM SPSS Modeler Text Analytics

SPSS Modeler Text Analytics offers powerful text analytic capabilities, which use advanced linguistic technologies and Natural Language Processing (NLP) to rapidly process a large variety of unstructured text data and, from this text, extract and organize the key concepts. Furthermore, SPSS Modeler Text Analytics can group these concepts into categories.

Around 80% of data held within an organization is in the form of text documents—for example, reports, Web pages, e-mails, and call center notes. Text is a key factor in enabling an organization to gain a better understanding of their customers' behavior. A system that incorporates NLP can intelligently extract concepts, including compound phrases. Moreover, knowledge of the underlying language allows classification of terms into related groups, such as products, organizations, or people, using meaning and context. As a result, you can quickly determine the relevance of the information to your needs. These extracted concepts and categories can be combined with existing structured data, such as demographics, and applied to modeling by using SPSS Modeler and its full suite of data mining tools to yield better and more-focused decisions.

Linguistic systems are knowledge sensitive—the more information contained in their dictionaries, the higher the quality of the results. SPSS Modeler Text Analytics is delivered with a set of linguistic resources, such as dictionaries for terms and synonyms, libraries, and templates. This product further allows you to develop and refine these linguistic resources to your context. Fine-tuning of the linguistic resources is often an iterative process and is necessary for accurate concept retrieval and categorization. Custom templates, libraries, and dictionaries for specific domains, such as CRM and genomics, are also included.

Installing IBM® SPSS® Modeler Premium Client

- [System requirements](#)
 - [Installing](#)
 - [After you install SPSS Modeler Premium](#)
 - [Removing IBM SPSS Modeler Premium](#)
-

System requirements

General requirements

IBM® SPSS® Modeler Premium must be installed on a system where SPSS Modeler Client is already installed.

To view the system requirements, go to <https://www.ibm.com/software/reports/compatibility/clarity/softwareReqsForProduct.html>.

SPSS Modeler Text Analytics requirements

Upgrading from earlier versions. Before installing SPSS Modeler Text Analytics version 18.4.0 you should save and export any TAPs, templates, and libraries from your current version that you want to use in the new version. We recommend that you save these files to a directory that will not get deleted or overwritten when you install the latest version.

After you install the latest version of SPSS Modeler Text Analytics you can load the saved TAP file, add any saved libraries, or import and load any saved templates to use them in the latest version.

Installing

Important: To install, you must be logged on to your computer with administrator privileges.

- [Installing from a downloaded file](#)
 - [Installing from a network location](#)
 - [Silent installation](#)
 - [Installing on Windows systems](#)
 - [Installing on UNIX systems](#)
 - [Silent installation](#)
-

Installing from a downloaded file

Windows XP

1. Double-click the file that you downloaded and extract all the files to some location on your computer.
2. Using Windows Explorer, browse to the location where you extracted the files and double-click *setup.exe*.
3. Follow the instructions that appear on the screen.

Windows Vista and later

Note: You must run the installer as administrator:

1. Double-click the file that you downloaded and extract all the files to some location on your computer.
 2. Using Windows Explorer, browse to the location where you extracted the files.
 3. Right-click *setup.exe* and choose Run as Administrator.
 4. Follow the instructions that appear on the screen.
-

Installing from a network location

1. Using Windows Explorer, browse to the location that your administrator provided for the *setup.exe* file.
 2. Right-click *setup.exe* and choose Run as Administrator.
 3. On the autoplay menu, click Install IBM® SPSS® Modeler Premium.
 4. Follow the instructions that appear on the screen.
-

Silent installation

Silent mode enables an installation to run on its own without any interaction; installing silently can free system administrators from the task of monitoring each installation and providing input to prompts and dialog boxes. This method is especially useful when you are installing SPSS® Modeler Premium on a number of different computers that have identical hardware.

Note: You must have administrator privileges to be able to run silent installations.

Windows - silent installation

You can complete a silent installation on Windows systems by using Microsoft Installer (MSI). Use *msiexec.exe* to install the MSI package.

The following options can be used:

Table 1. Silent installation options

Option	Description
<i>/i</i>	Specifies that the program is to install the product.
<i>/1*v</i>	Specifies verbose logging. For example, this form of log can be useful if you need to troubleshoot an installation.
<i>/qn</i>	Runs the installation without running the external user interface sequence.
<i>/s</i>	Specifies silent mode.

Option	Description
/v	Specifies that the Setup Program passes the parameter string to the call it makes to the MSI executable file (msiexec.exe). The following syntax requirements apply if you use this option: <ul style="list-style-type: none"> • You must place a backslash (\) in front of any quotation marks (" ") that are within existing quotation marks. • Do not include a space between the /v option and its arguments. • Multiple parameters that are entered with the /v option must be separated with a space. • To create a log file, specify the directory and file name at the end of the command. The directory must exist before you start the silent installation.
/x	Specifies that the program is to uninstall the product.

The following text shows an example of the MSI command:

```
c:>msiexec.exe /i ModelerPremium64.msi /qn /l*v
c:\temp\Modeler_Silent_Install.log
AgreeToLicense=true
```

Note: Depending on your system, you might need to change the ModelerPremium64.msi file in the preceding example.

Windows - silent uninstalling

The following text shows an example of the MSI command to silently uninstall the software:

```
C:>msiexec.exe /x ModelerPremium64.msi /qn /norestart
```

After you install SPSS Modeler Premium

SPSS Modeler Text Analytics data directory location

SPSS® Modeler Text Analytics will use the default installation locations to update and write files as necessary in the normal operation of SPSS Modeler Text Analytics .

On the SPSS Modeler Text Analytics client, data is written to a database that is installed to C:\ProgramData\IBM\SPSS\TextAnalytics\<version>\tmwb_<version>.db.

SPSS Modeler Text Analytics on Windows Vista

If you are installing SPSS Modeler Text Analytics on Windows Vista you must complete an additional step after you complete the installation.

Add modify permissions to the file: C:\ProgramData\IBM\SPSS\TextAnalytics\<version>\tmwb_<version>.db. This prevents various errors being created when trying to load templates or execute a text mining model builder.

Removing IBM SPSS Modeler Premium

To uninstall IBM® SPSS® Modeler Premium, perform the following steps:

1. From the Windows Start menu choose:
Settings > Control Panel
2. From the Control Panel, choose Add or Remove Programs.
3. Click Change or Remove Programs.
4. Select IBM SPSS Modeler Premium from the list of currently installed programs, and click Change/Remove. If you have more than one version installed on the computer, be sure to select the version that you want to remove.

A message will be displayed when the uninstallation process completes.

Installing IBM® SPSS® Modeler Premium Server

- [System requirements](#)
- [Installing](#)
- [Removing IBM SPSS Modeler Premium Server](#)

System requirements

IBM® SPSS® Modeler Premium Server must be installed on a system where SPSS Modeler Server is already installed. The requirements for IBM SPSS Modeler Premium Server are identical to those for SPSS Modeler Server.

Installing

Important: To install, you must be logged on to your computer with administrator privileges.

- [Installing from a downloaded file](#)
- [Installing from a network location](#)
- [Silent installation](#)
- [Installing on Windows systems](#)
- [Installing on UNIX systems](#)
- [Silent installation](#)

Installing on Windows systems

IBM® SPSS® Modeler Premium Server must be installed to the SPSS Modeler Server installation location. If SPSS Modeler Server is not installed, the IBM SPSS Modeler Premium Server installation will fail.

To install IBM SPSS Modeler Premium Server, perform the following steps.

1. Log on to the server computer with administrator privileges.
2. For your downloaded eAssembly:
 - Double-click the file and extract the installation files.
 - Go to the location where the installation files were extracted and double-click *Server64.exe*.
3. Follow the instructions that appear on the screen.
4. Restart the SPSS Modeler Server host when installation has completed.

Installing on UNIX systems

IBM® SPSS® Modeler Premium Server must be installed to the SPSS Modeler Server installation location. If SPSS Modeler Server is not installed, the IBM SPSS Modeler Premium Server installation will fail.

You must ensure that kernel limits on the system are sufficient for the operation of IBM SPSS Modeler Premium Server. We recommend that at least 4GB is available. Use the command **ulimit -a** to establish the existing size and increase it if required.

To install SPSS Modeler Premium Server, perform the following steps:

1. Ensure that SPSS Modeler is not running on the target machine.
2. Log in as the user that installed SPSS Modeler ServerProfessional, and ensure this user can execute the installer.
3. For your downloaded eAssembly:
 - Double-click the file and extract the installation files to a convenient location.
 - Change directories to the location where the installation files were extracted.

Note: If you are in a shell, instead of a user interface, extract the files by using the command `unzip <image name>.zip`.
4. Run the .bin file (for example; premium_server_aix64.bin or premium_server_zlinux64.bin).
5. Follow the displayed instructions. When prompted for an installation directory, use the SPSS Modeler Server installation directory. If you specify a different directory, an error message is displayed.
6. When installation has completed, restart the SPSS Modeler Server host.

Silent installation

Silent mode enables an installation to run on its own without any interaction; installing silently can free system administrators from the task of monitoring each installation and providing input to prompts and dialog boxes. This method is especially useful when you are installing SPSS® Modeler Premium on a number of different computers that have identical hardware.

Note: You must have administrator privileges to be able to run silent installations.

Windows - silent installation

You can complete a silent installation on Windows systems by using Microsoft Installer (MSI). Use msieexec.exe to install the MSI package.

The following options can be used:

Table 1. Silent installation options

Option	Description
/i	Specifies that the program is to install the product.
/l*v	Specifies verbose logging. For example, this form of log can be useful if you need to troubleshoot an installation.
/qn	Runs the installation without running the external user interface sequence.
/s	Specifies silent mode.
/v	Specifies that the Setup Program passes the parameter string to the call it makes to the MSI executable file (msiexec.exe). The following syntax requirements apply if you use this option: <ul style="list-style-type: none">• You must place a backslash (\) in front of any quotation marks (" ") that are within existing quotation marks.• Do not include a space between the /v option and its arguments.• Multiple parameters that are entered with the /v option must be separated with a space.• To create a log file, specify the directory and file name at the end of the command. The directory must exist before you start the silent installation.
/x	Specifies that the program is to uninstall the product.

The following text shows an example of the MSI command:

```
c:\>msiexec.exe /i ModelerPremiumServer64.msi /qn /L*v  
c:\temp\Modeler_Silent_Install.log  
AgreeToLicense=true
```

Note: Depending on your system, you might need to change the ModelerPremiumServer64.msi file in the preceding example.

Windows - silent uninstalling

The following text shows an example of the MSI command to silently uninstall the software:

```
c:\>msiexec.exe /x ModelerPremium64.msi /qn /norestart
```

Linux / UNIX - silent installation

To complete a silent installation on Linux or UNIX systems:

1. In the same location where you copied the installer files, create an installer.properties file.
2. In a text editor, set the installer.properties values. The following text shows an example of an installer.properties file:

```
=====  
# Thu Jan 29 11:35:37 GMT 2015  
# Replay feature output  
# -----  
# This file was built by the Replay feature of InstallAnywhere.  
# It contains variables that were set by Panels, Consoles or Custom Code.  
  
#Indicate whether the license agreement been accepted  
#-----  
LICENSE_ACCEPTED=TRUE  
  
#Choose Install Folder  
#-----  
USER_INSTALL_DIR=/usr/IBM/SPSS/ModelerServer/17.0  
  
#Install  
=====
```

3. Ensure that the value for **USER_INSTALL_DIR** matches your installation directory location. The directory path cannot contain spaces.
4. Save the file.
5. Run the installer by using the following command:

```
./<installer_name> -i silent -f installer.properties
```

Where <installer_name> is the installer .bin file.

Linux / UNIX - silent uninstalling

To silently uninstall the software, you can run the uninstaller in one of two ways:

- Execute the following command:

```
./<installer_path>/Uninstall_IBM_SPSS_MODELER_PREMIUM_SERVER/Uninstall_IBM_SPSS_MODELER_PREMIUM_SERVER  
-i silent
```

Where <installer_path> is the path name to the IBM® SPSS Modeler Server installation directory.

- Alternatively, if you have an installer.properties file, the following text shows an example of the command to silently uninstall the software:

```
./premium_server_linux64.bin -i silent -f ./installer.properties
```

Removing IBM® SPSS® Modeler Premium Server

- [Removing from Windows systems](#)
- [Removing from UNIX systems](#)

Removing from Windows systems

To uninstall IBM® SPSS® Modeler Premium Server, perform the following steps:

1. From the Windows Start menu choose:
Settings > Control Panel
2. From the Control Panel, choose Add or Remove Programs.
3. Click Change or Remove Programs.
4. Select IBM SPSS Modeler Premium Server from the list of currently installed programs, and click Change/Remove. If you have more than one version installed on the computer, be sure to select the version that you want to remove.

A message will be displayed when the uninstallation process completes.

Removing from UNIX systems

To uninstall IBM® SPSS® Modeler Premium Server, remove the program files and, if you have configured the system for automatic start up, disable automatic start up.

IBM® SPSS® Modeler - Essentials for R: Installation Instructions

- [Overview](#)
- [Install the IBM SPSS Modeler application](#)
- [Download and install R](#)
- [Download and install IBM SPSS Modeler - Essentials for R](#)
- [Repairing an installation](#)
- [Uninstalling IBM SPSS Modeler - Essentials for R components](#)

Overview

This document contains the instructions for installing IBM® SPSS® Modeler - Essentials for R.

IBM SPSS Modeler - Essentials for R provides you with tools you need to start using custom R scripts for model building and scoring within the Extension nodes in IBM SPSS Modeler. It includes the IBM SPSS Modeler - Integration Plug-in for R for IBM SPSS Modeler 18.4.0.

To use the R nodes in IBM SPSS Modeler Client, you must have the following components installed on the local machine:

- IBM SPSS Modeler 18.4.0. See the topic [Install the IBM SPSS Modeler application](#) for more information.

- R environment. See the topic [Download and install R](#) for more information.
- IBM SPSS Modeler - Essentials for R. See the topic [Download and install IBM SPSS Modeler - Essentials for R](#).

To use the R nodes with IBM SPSS Modeler Server, you must have the following components installed on the server machine:

- IBM SPSS Modeler Server 18.4.0. See the topic [Install the IBM SPSS Modeler application](#) for more information.
- R environment. See the topic [Download and install R](#) for more information.
- IBM SPSS Modeler - Essentials for R. See the topic [Download and install IBM SPSS Modeler - Essentials for R](#) for more information. The bit rate of IBM SPSS Modeler - Essentials for R that is installed must be the same as the installed version of IBM SPSS Modeler Server.

Notes:

- The Windows installer for IBM SPSS Modeler - Essentials for R is the same for both IBM SPSS Modeler and IBM SPSS Modeler Server. For example, the 64-bit installer for IBM SPSS Modeler - Essentials for R applies to both the 64-bit version of IBM SPSS Modeler and the 64-bit version of IBM SPSS Modeler Server.
- Starting with version 18.2.2, there's no longer a separate IBM SPSS Modeler - Essentials for R installer for Mac. It's part of the default SPSS Modeler installation, and the default R_HOME: path is /Library/Frameworks/R.framework/Resources. If you use a different path, you must edit the file config.ini located under /Applications/IBM/SPSS/Modeler/<version>/SPSSModeler.app/Contents/ext/bin/pasw.rstats and change config.ini to switch to your \${R_HOME} path on Mac.

Install the IBM SPSS Modeler application

There are no additional operating system and hardware requirements. The components that are installed with IBM® SPSS® Modeler - Essentials for R work with any valid IBM SPSS Modeler license.

If you have not already done so, follow the instructions that are provided with the software to install one of the IBM SPSS Modeler applications on the computer where you will install IBM SPSS Modeler - Essentials for R.

Note: If you are using Windows, and are installing IBM SPSS Modeler - Essentials for R on a desktop machine, you must also install IBM SPSS Modeler 18.4.0 on the desktop machine. If you are installing IBM SPSS Modeler - Essentials for R on a server machine, you must also install IBM SPSS Modeler Server 18.4.0 on the server machine.

Download and install R

Version 18.4.0 of IBM® SPSS® Modeler - Essentials for R requires an installation of R. Version 4.0.4 is recommended. Install R on the computer where you will install IBM SPSS Modeler - Essentials for R.

Prerequisites

The target computer where you will install Essentials for R must have X11. If the target computer has a physical display, then it most likely has X11. The steps that follow describe the process for installing X11, if necessary.

1. Install the X11 client and server

- For Linux distributions that use `yum`, install the X11 client and server software with:

```
yum groupinstall "X Window System" "Desktop" "Fonts" "General Purpose Desktop"
yum update xorg-x11-server-Xorg
yum install xorg-x11-server-Xvfb.x86_64
```

- For Linux distributions that use `apt-get`, install the X11 client and server software with:

```
apt-get install xorg xterm
apt-get install xserver-xorg xserver-xorg-core xserver-xorg-dev
apt-get install xvfb
```

2. Install openGL

- For Linux distributions that use `yum`, install openGL with:

```
yum install mesa-libGL-devel mesa-libGLU-devel libpng-devel
```

- For Linux distributions that use `apt-get`, install openGL with:

```
apt-get install libgl1-mesa-glx libgl1-mesa-dev libglu1-mesa libglu1-mesa-dev
```

3. Start Xvfb. For more information, see <http://www.x.org/archive/X11R7.6/doc/man/man1/Xvfb.1.xhtml>.

4. Set the `DISPLAY` environment variable. The general form for the `DISPLAY` variable is:

```
export DISPLAY=<Hostname>:<D>.<S>
```

In the preceding statement,

<Hostname> is the name of the computer that hosts the X display server. To specify localhost, omit the value of <Hostname>. <D> is the display number of the Xvfb instance. <s> is the screen number, which is typically 0.

Note: The DISPLAY environment variable must be set before you start the IBM SPSS Modeler server.

5. 4.0.4 is the recommended R version. Note that the versions of zlib, bzip2, xz, and pcre that were included with old versions of R have been removed. So if you choose to install R from source, you must install the dependent packages zlib, bzip2, xz, pcre, and curl. You must also set the shared library path by adding the following line to the .bash_profile for the user who runs R or /usr/local/lib/etc/ld.so.conf.

```
export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
```

For more information, see the *R Installation and Administration* manual at <https://www.r-project.org/>.

- For Linux distributions that use yum, install packages with:
 - yum install zlib zlib-devel
 - yum install bzip2 bzip2-devel
 - yum install xz xz-devel
 - yum install pcre pcre-devel
 - yum install libcurl libcurl-devel
- For Linux distributions that use apt-get, install packages with:
 - apt-get install zlib1g zlib1g-dev
 - apt-get install bzip2 bzip2-dev libbz2-dev
 - apt-get install liblzma-dev
 - apt-get install libpcre3 libpcre3-dev

Note that for libcurl, you can install one of them:

- apt-get install libcurl4-openssl-dev
- apt-get install libcurl4-gnutls-dev
- apt-get install libcurl4-nss-dev

In addition to installing X11, we also recommend installing tcl/tk before installing R.

Installing R from a package manager

Your distribution's repository may include R. If so, you can install R using your distribution's standard package manager (such as the RPM Package Manager or the Synaptic Package Manager).

- For Linux distributions that use yum, you can install R with `yum install R`.
- For Linux distributions that use apt-get, you can install R from the command:

```
apt-get install r-base=<Version> r-base-core=<Version> r-base-dev=<Version>
```

where <Version> is the name of the version. Note that you might need to update the file /etc/apt/source.list to add new sources.

Building and installing R from source

The source for R is available from <ftp://ftp.stat.math.ethz.ch/Software/CRAN/src/base/R-3/>.

1. Create a temporary directory where you will uncompress and unpack the R source. For example, at a command prompt type:
`mkdir ~/Rsource`

2. Download the source code for building R, for example *R-4.0.4.tar.gz*, and save it to the temporary directory.

3. Change to the temporary directory. For example, at a command prompt type:

```
cd  
~/Rsource
```

4. Uncompress and unpack the R source to the temporary directory. For example, at a command prompt type:
`tar xzf R-4.0.4.tar.gz`

5. Change to the source directory. For example, at a command prompt type:

```
cd  
R-4.0.4
```

Note: To install R to the default directory, you must run the following step as root, either by logging in as root or using the `sudo` command. It is recommended that you read the information in *doc/html/R-admin.html* (located under the directory where you unpacked the R source) before proceeding with configuring, building and installing R.

6. Run the following commands to specify necessary compiler settings (see the special settings for PowerLinux):

```
export CC="gcc -m64"  
export CXXFLAGS="-m64 -O2 -g"  
export FFLAGS="-m64 -O2 -g"  
export FCFLAGS="-m64 -O2 -g"  
export LDFLAGS="-L/usr/local/lib64"  
export LIBnn=lib
```

PowerLinux settings:

```
export CC=<XLC_PATH>/bin/xlc_r -q64"
export CFLAGS="-g -O2 -qstrict -qfloat=nomaf:fenv"
export F77=<XLF_PATH>/xlf_r -q64"
export FFLAGS="-g -O3 -qstrict -qfloat=nomaf:fenv -qextname"
export CXX=<XLC_PATH>/bin/xlc_r -q64"
export CPICFLAGS=-qpic
export CXXPICFLAGS=-qpic
export FPICFLAGS=-qpic
export SHLIB_LDFLAGS=-qmksrhobj
export SHLIB_CXXLDFLAGS=-G
export FC=<XLF_PATH>/xlf_r -q64"
export FCFLAGS="-g -O3 -qstrict -qfloat=nomaf:fenv -qextname"
export FCPICFLAGS=-qpic
export CXX1XSTD=-qlanglvl=extended0x
```

Where <XLC_PATH> and <XLF_PATH> are the locations of IBM XL C/C++ for Linux and IBM XL Fortran for Linux respectively.

If installing R on PowerLinux P8LE servers (Ubuntu or RedHat), you must install the XL C/C++ Compiler and the XL Fortran Compiler (16.1.1.12 or later).

Also for PowerLinux, run these commands:

```
xlf -qpreprocess -qnoobject -d src/modules/lapack/dlapack.f
mv -f Fdlapack.f src/modules/lapack/dlapack.f
xlf -qpreprocess -qnoobject -d src/extrablas/blas.f
mv -f Fblas.f src/extrablas/blas.f
```

7. Configure, build, and install R. Be sure to configure R with the `--enable-R-shlib` and `--with-x` arguments. For example, at a command prompt type (see the special settings for PowerLinux):

```
./configure --enable-R-shlib --with-x && make && make install
```

PowerLinux settings:

```
./configure --enable-R-shlib --with-x --with-readline=no --disable-openmp&& gmake
&& gmake install
```

For details about building R on IBM z Systems, see <https://github.com/linux-on-ibm-z/docs/wiki/Building-R>.

Note: The readline entry is optional, depending on how your system is configured.

Download and install IBM SPSS Modeler - Essentials for R

Be sure to use a version of IBM® SPSS® Modeler - Essentials for R that is compatible with the version of IBM SPSS Modeler on your machine. Within a major version of IBM SPSS Modeler, such as 18.0, you must use a version of IBM SPSS Modeler - Essentials for R that has the same major version.

For users who are working in distributed mode (with IBM SPSS Modeler Server) please install IBM SPSS Modeler - Essentials for R on the server machine.

Log on to Passport Advantage and download version 18.4.0 of IBM SPSS Modeler - Essentials for R. Be sure to download the version of IBM SPSS Modeler - Essentials for R for the operating system of your IBM SPSS Modeler application.

Tip: After installing IBM SPSS Modeler - Essentials for R, if you ever need to know what version is installed, you can run the following command in the R console.

```
packageVersion("ibmspsscf92")
```

- [Install IBM SPSS Modeler - Essentials for R for Windows](#)
- [Install IBM SPSS Modeler - Essentials for R for UNIX](#)
- [IBM SPSS Modeler - Essentials for R for Mac](#)
- [Silent installation](#)
- [Running Extension nodes in IBM SPSS Modeler Solution Publisher and IBM SPSS Collaboration and Deployment Services](#)

Install IBM SPSS Modeler - Essentials for R for Windows

Windows Vista, Windows 7, or Windows Server 2008

You must run the installer as administrator:

1. Using Windows Explorer, browse to the folder where you downloaded the file.
2. Right-click the downloaded file and choose Run as Administrator.
3. Follow the instructions that are displayed on the screen.

Pushing an installation

As an alternative to the manual installation described above, you can push the installation to Windows computers. This is most useful for network administrators who need to install to multiple end users. Following is the form of the command line for pushing an installation:

```
<installer_name> -i silent
```

Here, `<installer_name>` is the name of the installer file for IBM® SPSS® Modeler - Essentials for R, for example:
`SPSS_Modeler_ESSENTIALS_<version>_win64.exe`.

Increasing the memory limit

Under Windows, R imposes a limit on the total memory allocation that is available to an R executable session. This limit restricts the embedded R process `r_start.exe`.

If required, you can modify the numeric value to increase the memory limit; to do this, add an option in the end of the `C:\Program Files\IBM\SPSS\Modeler\<version>\ext\bin\pasw.rstats\config.ini` file. For example, to raise the limit to 4096Mb:

```
Max_Mem_Size=4096
```

Install IBM SPSS Modeler - Essentials for R for UNIX

1. Start a terminal application.
2. Change to the directory where you downloaded IBM® SPSS® Modeler - Essentials for R. At the command prompt, type:
`./<<filename>>`

where `<<filename>>` is the name of the file you downloaded. You must ensure that this file has execute permission before you attempt to run the command.

Note: You must run the previous command as root, either by logging in as root or (if installing as non-root) by using the `sudo` command and having write permission to `<SPSS Modeler installation directory>/ext/bin` and `<USER_R_HOME>`. In addition, you need to install the `gcc` and `gfortran` compilers before you install IBM SPSS Modeler - Essentials for R.

3. Follow the instructions that are displayed on the screen. When prompted for the location of R, you can obtain the R home directory by running `R.home()` from the R prompt.

Note: To ensure that SPSS Modeler can launch R successfully, export the library search paths that are required by `libR.so` to the `DLLIBPATH` variable in the `modelersrv.sh` file in the SPSS Modeler Server installation directory. To find all the `libR.so` libraries that are referenced, use the command `ldd <R_HOME>/lib/libR.so`.

IBM SPSS Modeler - Essentials for R for Mac

Starting with version 18.2.2, there's no longer a separate IBM® SPSS® Modeler - Essentials for R installer for Mac. It's part of the default SPSS Modeler installation, and the default `R_HOME`: path is `/Library/Frameworks/R.framework/Resources`. If you use a different path, you must edit the file `config.ini` located under `/Applications/IBM/SPSS/Modeler/<version>/SPSSModeler.app/Contents/ext/bin/pasw.rstats` and change `config.ini` to switch to your `${R_HOME}` path on Mac.

Silent installation

As an alternative to the manual installations described previously, you can also run a silent installation. This is most useful for network administrators who need to install to multiple end users. To run a silent installation, do the following:

1. Start a terminal application.
2. Change to the directory where you downloaded IBM® SPSS® Modeler - Essentials for R.
3. Using a text editor, create a response file named `install.properties`.
4. Add the following properties and associated values to the response file:

```
USER_INSTALL_DIR=<R 4.0.4 home directory>
FRONTEND_INSTALL_DIR=<IBM SPSS Modeler location>/ext/bin
```

where <R 4.0.4 home directory> is the installation location of R 4.0.4 and <IBM SPSS Modeler location> is the installation location of IBM SPSS Modeler. For example, on UNIX:

```
USER_INSTALL_DIR=/usr/local/lib/R  
FRONTEND_INSTALL_DIR=/usr/IBM/SPSS/ModelerServer/19/ext/bin
```

For example, on Windows:

```
USER_INSTALL_DIR=C:\\Program Files\\R\\R-4.0.4  
FRONTEND_INSTALL_DIR=C:\\Program Files\\IBM\\SPSS\\Modeler\\18.4.0\\ext\\bin
```

5. Save install.properties to the directory that contains the .bin file for IBM SPSS Modeler - Essentials for R and change to that directory.
6. On UNIX, run the installer with the following command:

```
./<installer_name> -i silent
```

where <installer_name> is the name of the .bin file for IBM SPSS Modeler - Essentials for R. Note that you must run the previous command as root, either by logging in as root or using the **sudo** command.

On Windows, run the installer with the following command:

```
<installer_name> -i silent
```

where <installer_name> is the name of the installer file for IBM SPSS Modeler - Essentials for R, for example **SPSS_Modeler_REssentials_<version>_win32.exe**.

Alternatively, on UNIX, you can run the installer with the following command:

```
./<installer_name> -f <Response file location>
```

On Windows, you can run the installer with the following command:

```
<installer_name> -f <Response file location>
```

In both cases, <Response file location> is the file path to the response file. If you use this alternative command, you must add the following property to the response file:

```
INSTALLER_UI=[swing | console | silent]
```

Note: To use a different response file (other than install.properties), on UNIX run the installer with the following command:

```
./<installer_name> -i silent -f <response file name>
```

On Windows, run the installer with the following command:

```
<installer_name> -i silent -f <response file name>
```

Running Extension nodes in IBM SPSS Modeler Solution Publisher and IBM SPSS Collaboration and Deployment Services

If you want to run Extension nodes (formerly R nodes) in SPSS® Modeler Solution Publisher and run the Scoring Service on the IBM® SPSS Collaboration and Deployment Services server, you must install IBM SPSS Modeler - Essentials for R and the R environment with SPSS Modeler Solution Publisher and the IBM SPSS Collaboration and Deployment Services server.

Running Extension nodes (Extension Export node, Extension Output node, Extension Model node, Extension Transform node, and Extension Import node)

1. For the Extension nodes to work with SPSS Modeler Solution Publisher, install IBM SPSS Modeler - Essentials for R and the R environment on the same machine as the IBM SPSS Collaboration and Deployment Services server. During IBM SPSS Modeler - Essentials for R installation, point to the R environment installation directory and the SPSS Modeler Solution Publisher installation directory.
2. To run the Scoring Service on the IBM SPSS Collaboration and Deployment Services server, you must also install IBM SPSS Modeler - Essentials for R and the R environment on the same machine as the IBM SPSS Collaboration and Deployment Services server. During IBM SPSS Modeler - Essentials for R installation, point to the R environment installation directory and the local IBM SPSS Modeler Server location under the IBM SPSS Collaboration and Deployment Services server installation directory.
3. For R in CDB node execution, after setting up the environment as described in the previous steps, you must also set an environment variable as follows:
 - a. On the IBM SPSS Collaboration and Deployment Services server machine and the IBM SPSS Modeler client machine, create a system environment variable called **IBM_SPSS_MODELER_EXTENSION_PATH** that points to the folder that contains the R CDB node .cdf and .cfe files.
 - b. Make sure both IBM SPSS Collaboration and Deployment Services server and IBM SPSS Modeler client can access this path.

c. Restart IBM SPSS Collaboration and Deployment Services server and IBM SPSS Modeler client.

Note: To ensure that R can launch successfully, export the library search paths that are required by libR.so to the **DLLIBPATH** variable in the modelersrv.sh file in the IBM SPSS Modeler Solution Publisher installation directory. To find all the libR.so libraries that are referenced, use the command **ldd <R_HOME>/lib/libR.so**.

Repairing an installation

If you uninstall and then reinstall the IBM® SPSS® Modeler 18.4.0 application or your R environment, then you must also uninstall and then reinstall version 18.4.0 of IBM SPSS Modeler - Essentials for R.

Uninstalling IBM® SPSS® Modeler - Essentials for R components

- [Windows](#)
 - [UNIX](#)
-

Windows

Remove the following folder and files:

- ibmspsscf84 from <R 4.0.x home directory>\\library
 - config.ini from <IBM® SPSS® Modeler location>\\ext\\bin\\pasw.rstats
 - embeded.dll from <IBM SPSS Modeler location>\\ext\\bin\\pasw.rstats
-

UNIX

Remove the following folder and files:

- ibmspsscf84 from <R home directory>/library
 - config.ini from <IBM® SPSS® Modeler location>/ext/bin/pasw.rstats
 - libembeded.so from <IBM SPSS Modeler location>/ext/bin/pasw.rstats
-

Installing IBM SPSS Modeler Server Adapter

- [About IBM SPSS Modeler Server Adapter installation](#)

- [System requirements](#)

- [Installation files](#)

Before installing, you must obtain the installation files.

- [Getting started with Installation Manager](#)

Installing, updating, or uninstalling the product can be performed by using IBM® Installation Manager in wizard, console, or silent mode. However, you must configure an IBM Installation Manager repository or Passport Advantage preferences before performing these tasks.

- [Installing IBM SPSS Modeler Adapter](#)

You can install IBM SPSS® Modeler Adapter in wizard, console, or silent mode.

- [Configuring the adapter for IBM SPSS Collaboration and Deployment Services Web services on Linux](#)

- [Configuring the adapter for SPSS Statistics](#)

- [Troubleshooting](#)
-

About IBM SPSS Modeler Server Adapter installation

This guide provides installation instructions and information relating to the products available on the IBM® SPSS® Modeler Server Adapter installation media.

The adapters enable IBM SPSS Modeler and IBM SPSS Modeler Server to interact with an IBM SPSS Collaboration and Deployment Services repository. For more information, see the *IBM SPSS Modeler User's Guide*.

System requirements

Before you can install the adapter, you must be able to connect to a fully-functioning IBM® SPSS® Collaboration and Deployment Services repository. System requirements for this are given in the *Installation and Configuration Guide* for the appropriate repository version. Kerberos credentials are supported for running jobs and managing user roles.

IBM Installation Manager **1.9.1** must be installed on the machine where the adapter will be installed, and it must be configured to use the installation files. See the following section.

Note: Installing and running the adapter will consume additional resources of the repository host, most importantly memory. It is recommended that, prior to installation, you consult the application server documentation to confirm you have sufficient memory for your selected application server.

The repository server needs to have a valid Java Runtime Environment (JRE) set up in the PATH or JAVA_HOME environment variables, otherwise installation will fail.

Make sure you have sufficient free disk space before starting installation. A minimum of 10 GB is recommended. This installation is for the **8.x** version of the adapters and uses IBM Installation Manager to install all the adapters at the same time, so the installation may take longer than in previous releases.

If not present already, the Microsoft Visual C++ 2010 runtime library will be installed on the machine where the adapter will be installed.

Installation files

Before installing, you must obtain the installation files.

To obtain the installation files do one of the following:

- Download the files from the Passport Advantage site, and use local installation: Licensed customers with a Passport Advantage ID and password can download the necessary product repositories from the Passport Advantage site.
- Access the live repositories, and use web-based installation: If you have a Passport Advantage ID and password, you can use Installation Manager to install the product directly from IBM hosted repositories.

Getting started with Installation Manager

Installing, updating, or uninstalling the product can be performed by using IBM® Installation Manager in wizard, console, or silent mode. However, you must configure an IBM Installation Manager repository or Passport Advantage preferences before performing these tasks.

For complete information about Installation Manager, see the [IBM Installation Manager documentation](#).

Wizard mode

In wizard mode, you run Installation Manager from a graphical user interface.

Most of the time, you start Installation Manager with the default shortcuts that are installed with your version of Installation Manager.

From the installation location of Installation Manager, you can run the **IBMMIM** application file to start in wizard mode manually.

The default **IBMMIM** location for the operating system varies depending on the installation type (administrator, nonadministrator, or group).

Table 1. Default installation locations for IBMMIM

Operating system	Administrator	Nonadministrator	Group
Windows 2008 and Windows 2012	C:\Program Files [(x86)]\IBM\InstallationManager\IBMMIM.exe	C:\Users\user\IBM\InstallationManager\IBMMIM.exe	
Linux® and UNIX	/opt/IBM/InstallationManager/eclipse/IBMMIM	/user_home_directory/IBM/InstallationManager/eclipse/IBMMIM	/user_home_directory/IBM/InstallationManager_Group/eclipse/IBMMIM

Console mode

Use console mode when you do not have a graphics display device available or when you want to run Installation Manager without the graphical user interface. Installation Manager supports installing in an ASCII text-based mode that is called the console mode. Console mode is interactive text-based user interface to Installation Manager. For example, use console mode for server-side deployments when no graphical user interface is present, or for running the installation from a remote host.

To start console mode:

1. Open a command line.
2. Go to the `tools` subdirectory.
3. Run the command that is appropriate for the operating system:
 - Windows: `imcl.exe -c`
 - Linux, UNIX: `./imcl -c`

The default `tools` location varies depending on the operating system and installation type (administrator, nonadministrator, or group). For more information, see the Installation Manager documentation.

Table 2. Default installation locations for the `tools` subdirectory

Operating system	Administrator	Nonadministrator	Group
Windows 2003, Windows 2008 and Windows 2012	<code>C:\Program Files [(x86)]\IBM\InstallationManager\eclipse\tools</code>	<code>C:\Users\user\IBM\InstallationManager\eclipse\tools</code>	
Linux and UNIX	<code>/opt/IBM/InstallationManager/eclipse/tools</code>	<code>/user_home_directory/IBM/InstallationManager/eclipse/tools</code>	<code>/user_home_directory/IBM/InstallationManager_Group/eclipse/tools</code>

Silent mode

Use silent installations to deploy software to multiple systems, or to an enterprise. Silent installations are defined by a response file and started from the command line or a batch file. The response file is provided with the product distribution. For more information, see [Installing silently by using a response file](#).

- [Repository preferences](#)
An IBM Installation Manager repository is a location that stores data for installing, modifying, rolling back, or updating packages.
- [Passport Advantage preferences](#)
IBM Installation Manager can access installation packages from Passport Advantage®. Passport Advantage is a centralized online location for the acquisition of IBM software offerings.

Repository preferences

An IBM Installation Manager repository is a location that stores data for installing, modifying, rolling back, or updating packages.

Before you install, modify, or update packages, obtain the installation repository location from your administrator or from IBM.

Note: To successfully access an installation repository, the repository location path must not contain an ampersand (&).

The following topics provide instructions for setting repository preferences in wizard and console mode.

- [Setting repository preferences in wizard mode](#)
You can add, edit, or remove repositories and modify the repository order in the repository table using wizard mode.
- [Setting repository preferences in console mode](#)
You can use console mode to add, remove, open, move, or close repositories.

Setting repository preferences in wizard mode

You can add, edit, or remove repositories and modify the repository order in the repository table using wizard mode.

About this task

You can clear credentials for a repository or test a connection to a repository. You might find both a `diskTag.inf` file and a `repository.config` file in the IBM® product installation files. Use the `diskTag.inf` file when you select a repository location.

Procedure

To add, edit, or remove a repository location:

1. Start Installation Manager in wizard mode by using **IBMMIM**. For more information, see [Getting started with Installation Manager](#).
2. Click **File > Preferences > Repositories**.
The Repositories page opens and shows available repositories, repository locations, and the connection status for the repositories.
3. Click **Add Repository**.
4. Enter the repository location or click **Browse**. When you browse, go to the repository location and select the **diskTag.inf** file, the **repository.config** file, the **.zip** file, or the **.jar** file as appropriate for your environment.
5. Click **OK**.
If you provided an HTTPS or restricted FTP repository location, you are prompted to enter a user ID and password.
The new repository location is added to the list. If the repository is not connected, a red box shows in the Connection column.
6. Optional: Select Search service repositories during installation and updates. Installation Manager searches the service repositories at IBM.com for updates to installed packages.
7. Click **OK** to close the Preference page.

Setting repository preferences in console mode

You can use console mode to add, remove, open, move, or close repositories.

About this task

A selected option is indicated by an **x** in brackets: **[x]**. Options that are not selected are indicated by empty brackets: **[]**. You can press **Enter** to select the default entry or select a different command. For example, **[N]** indicates that the default selection is **N** for the **Next** command.

Procedure

To add a repository:

1. Start Installation Manager in console mode by using **imcl -c**. For more information, see [Getting started with Installation Manager](#).
2. Enter **P: Preferences**.
3. Enter **1: Repositories**.
4. Enter **D: Add repository**.
5. Enter a repository location such as **C:\installation_files\repository.config**.
If you add a repository that requires credentials, you are prompted to provide the required credentials.
Use the correct case when you enter the repository location. If the correct case is not used, the package is not shown in the list of available packages for installation.
 - a. Enter **P: Provide credentials and connect**.
 - b. Enter the **user_name** and press **Enter**.
 - c. Enter the **password** and press **Enter**.
 - d. Enter **1** to save the password.
 - e. Enter **O: Ok**.
6. Enter **A: Apply Changes and Return to Preferences Menu**.
7. Enter **R: Return to Main Menu**.

Passport Advantage preferences

IBM® Installation Manager can access installation packages from Passport Advantage®. Passport Advantage is a centralized online location for the acquisition of IBM software offerings.

Before you install, modify, or update packages, obtain valid Passport Advantage credentials.

The following topics provide instructions for setting Passport Advantage preferences in wizard and console mode.

- [**Setting Passport Advantage preferences in wizard mode**](#)
You can set the Installation Manager Passport Advantage preferences to connect to Passport Advantage using wizard mode.
- [**Setting Passport Advantage preferences in console mode**](#)
You can set the Installation Manager Passport Advantage preference to connect to Passport Advantage in console mode.

Setting Passport Advantage preferences in wizard mode

You can set the Installation Manager Passport Advantage® preferences to connect to Passport Advantage using wizard mode.

About this task

Important: If you share an instance of Installation Manager with other users, see the [Installation Manager documentation](#) for information on installing as an administrator, nonadministrator, or group.

Procedure

To set Passport Advantage preferences:

1. Start Installation Manager in wizard mode by using **IBMMIM**. For more information, see [Getting started with Installation Manager](#).
2. Click **File > Preferences > Passport Advantage**.
3. Select the **Connect to Passport Advantage** check box to connect to the Passport Advantage repository.
The **Password Required** window opens.
4. Enter a user name and password for Passport Advantage.
5. Optional: Select **Save password** to save the user name and password credentials.
If you do not save the user name and password credentials, you are prompted for these credentials each time you access Passport Advantage.
6. Click **OK** to close the **Password Required** window.
7. Click **OK** to close the **Preferences** window.

What to do next

To delete saved user name and password credentials:

1. Click **File > Preferences > Passport Advantage**.
2. Click **Clear Credentials**.
3. Click **OK** in the **Confirm Clear Credentials** window.

Setting Passport Advantage preferences in console mode

You can set the Installation Manager Passport Advantage® preference to connect to Passport Advantage in console mode.

Procedure

1. Start Installation Manager in console mode by using **imcl -c**. For more information, see [Getting started with Installation Manager](#).
2. Enter **P: Preferences**.
3. Enter **6: Passport Advantage**.
4. Enter **1: Connect to Passport Advantage**.
A selected option is indicated by an **x** in brackets: **[x]**.
5. Enter **P: Provide credentials and connect**.
6. Enter the user name for the Passport Advantage account.
7. Enter the password.
If you do not save the user name and password credentials, you are prompted for these credentials each time you access Passport Advantage.
 - a. Optional: If you entered a password, enter **1: Save password if valid**.
8. Enter **O: OK** to save the credentials.

Installing IBM SPSS Modeler Adapter

You can install IBM SPSS Modeler Adapter in wizard, console, or silent mode.

You must shut down the IBM SPSS Collaboration and Deployment Services server before starting the installation, and ensure that the application server is in the following state:

- **IBM WebSphere standalone:** Server must be stopped.
- **IBM WebSphere managed:** Managed server must be stopped, Deployment Manager server must be running.
- **IBM WebSphere cluster:** Cluster members must be stopped, Deployment Manager server must be running.
- **JBoss:** Server must be stopped.

- **Oracle WebLogic standalone:** Server must be stopped.
- **Oracle WebLogic managed:** Managed server must be stopped, WebLogic administration server must be running.
- **Oracle WebLogic cluster:** Cluster members must be stopped, WebLogic administration server must be running.

- **Installing in wizard mode**

You can install IBM SPSS Modeler Adapter by using IBM Installation Manager in wizard mode.

- **Installing in console mode**

You can install IBM SPSS Modeler Adapter by using IBM Installation Manager in console mode.

- **Installing silently by using a response file**

You can use a response file to install in silent mode.

Installing in wizard mode

You can install IBM® SPSS® Modeler Adapter by using IBM Installation Manager in wizard mode.

Before you begin

Before you can install, IBM Installation Manager must have access to the repository that contains the package. You must also shut down the application server and the IBM SPSS Collaboration and Deployment Services server before starting the installation.

- If you have an IBM Passport Advantage® account, you can install packages from the Passport Advantage site. For more information about connecting to a Passport Advantage repository, see [Setting Passport Advantage preferences in wizard mode](#).
- If you are installing from a repository that is not on the Passport Advantage site, you must specify the repository in the preferences before you install. For more information, see [Setting repository preferences in wizard mode](#).

Procedure

To install IBM SPSS Modeler Adapter:

1. Start Installation Manager in wizard mode by using [IBMIM](#). For more information, see [Getting started with Installation Manager](#).
2. In Installation Manager, click Install.
Installation Manager searches the defined repositories for available packages. If no available packages are found, verify that you specified the repository correctly. See [Setting repository preferences in wizard mode](#).
3. If a new version of Installation Manager is found, you might be prompted to confirm the installation. Click Yes to proceed. Installation Manager automatically installs the new version, restarts, and resumes.
4. The Install page of Installation Manager lists all the packages that were found in the repositories that Installation Manager searched. Only the most recent version of the package is shown. To show all versions of a package that Installation Manager finds, select Show all versions.
Click a package version to show the package description in the Details pane. If more information about the package is available, a More info link is included at the end of the description text.
If you are running Installation Manager in group mode, you can install only packages that are enabled for installing in group mode. If the package is not enabled for installing in group mode, you receive an error and cannot continue with the package installation in group mode.
5. Select the IBM SPSS Modeler Adapter package. Click Next.
6. On the Licenses page, read the license agreements for the selected package. After you accept the license agreement, click Next to continue.
7. On the Location page, enter the path for the shared resources directory in the Shared Resources Directory field. The shared resources directory contains resources that can be shared by multiple package groups. Click Next.
Important: You can specify the shared resources directory only the first time that you install a package. Select the drive with enough available space to ensure adequate space for the shared resources of future packages. You cannot change the location of the shared resources directory unless you uninstall all packages.
8. On the Location page, either choose a package group into which to install the packages or create a package group.
A package group is a directory that contains resources that packages share with other packages in the same group. The first time that you install a package, you must create a package group. If you select more than one package to install, verify that the packages can be installed in the same package group by checking the documentation for the packages. For packages that cannot be installed in the same package group, install one package in one package group. After the installation completes, install the second package in a different package group.

Option	Description
Use the existing package group	Select a package group into which to install the packages. If the packages being installed are not compatible with the selected group, an alert reports the conflict. If a group is not compatible, either select a different group or create a new group.
Create a new package group	Click Browse to specify the installation directory for the packages. If you are installing on a 64-bit operating system, select the architecture for the installation as either 32-bit or 64-bit.

Click Next to continue the installation.

9. On the next Location page, select the translations to install for packages in the package group. The corresponding language translations for the graphical user interface and documentation are installed. Choices apply to all packages that are installed in this package group. This option may not apply to all product installations. Click Next to continue.
10. On the Features page, select the package features to install.
 - a. Optional: To see the dependency relationships between features, select Show Dependencies.
 - b. Optional: Click a feature to view its brief description under Details.
 - c. Select or clear features in the packages. Installation Manager automatically enforces dependencies with other features and shows updated download size and disk space requirements for the installation.

To restore the default features that are selected for the packages, click Restore Default.
11. When you are finished selecting features, click Next.
12. On the Summary page, review your choices before you install the packages.
On Windows, Installation Manager checks for running processes. If processes are blocking the installation, a list of these processes is shown in the Blocking Processes section. You must stop these processes before you continue the installation. Click Stop All Blocking Processes. If there are no processes that must be stopped, you do not see this list. The running processes lock files that must be accessed or modified by Installation Manager.
13. Click Install.

When the installation process completes, you receive a confirmation message.

Installing in console mode

You can install IBM® SPSS® Modeler Adapter by using IBM Installation Manager in console mode.

Before you begin

Before you can install, Installation Manager must have access to the repository that contains the package. You must also shut down the application server and the IBM SPSS Collaboration and Deployment Services server before starting the installation.

- If you have an IBM Passport Advantage® account, you can install packages from the Passport Advantage site. For more information about connecting to a Passport Advantage repository, see [Setting Passport Advantage preferences in console mode](#).
- If you are installing from a repository that is not on the Passport Advantage site, you must specify the repository in the preferences before you install. For more information, see [Setting repository preferences in console mode](#).

Procedure

To install in console mode:

1. Start Installation Manager in console mode by using `imcl -c`.
The default `imcl` location varies depending on the operating system and installation type (administrator, nonadministrator, or group). For more information, see [Getting started with Installation Manager](#).
2. Enter 1: Install - Install software packages.
Packages that can be installed are listed.
If you have repositories that require credentials and you did not save the credentials, you are prompted to provide these credentials
3. Enter 1: [] *package_name*.
To select a package, enter the number that is next to the package. This example selects the first package listed. If the selected package requires a later version of Installation Manager, you are prompted to install the later version.
4. On the Select screen, enter the number that is next to the package that you want to install.
 - 1: Choose version *package_version* for installation. This option shows when you chose a package that is not selected for installation.
The 1: Do NOT install version *package_version* option shows when you chose a package that is selected for installation.
 - 2: Show all available versions of the package.
5. Optional: Enter 0: Check for Other Versions, Fixes, and Extensions. Installation Manager searches available repositories for other versions, fixes, or extensions of the selected package.
 - For Installation Manager to search the default repository for the installed packages, the Search service repositories during installation and updates preference must be selected. This preference is selected by default. To access this preference, go to the Repositories preference page.
 - Typically, Internet access is required.
 - The Check for Other Versions, Fixes, and Extensions option indicates the number of other versions, fixes, or extensions found but does not list the found items. To see available versions, enter the number that is next to the package then enter 2: Show all available versions of the package.
6. Enter N: Next
7. Options for the Licenses screen:
 - 1: *product_name* - License Agreement. To view a license agreement, enter the number that is next to the product name. This example selects the first license agreement listed.

- A: [] I accept the terms in the license agreement.
 - D: [] I do not accept the terms in the license agreement. If you decline the license agreement, the installation is stopped. To continue the installation, you must accept the license agreement.
 - a. Enter A to accept the license agreement.
 - b. Enter N: Next.
8. To enter a different value for the shared resources directory, enter M: Shared Resources Directory. To accept the default value for shared resources directory or to continue after you enter a different value, enter N: Next.
Important: You can specify the shared resources directory only the first time that you install a package. Select the drive with enough available space to ensure adequate space for the shared resources of future packages. You cannot change the location of the shared resources directory unless you uninstall all packages.
9. Optional: To enter a different value for the package group location, enter M: Change Location. To accept the default values or to continue after you enter a different value, enter N: Next.
A package group is a directory that contains resources that packages share with other packages in the same group. The first time that you install a package, you must create a package group. If you select more than one package to install, verify that the packages can be installed in the same package group by checking the documentation for the packages. For packages that cannot be installed in the same package group, install one package in one package group. After the installation completes, install the second package in a different package group.
10. Enter the number that is next to the language to add or remove the language from the list of languages that are installed. You can select only one language at a time. Enter N: Next.
English is selected by default. You cannot clear the selection for the English language. Your language choices apply to all packages that are installed in the package group.
11. Enter the number that is next to the feature to add or remove the feature from the list of features that are installed. Enter N: Next.
This screen is not shown when your product does not have any features.
12. On the Summary screen, review your selections before you install the package.
Optional: To generate a response file, enter G: Generate an installation response file. Enter the name of the response file and use .xml as the file extension. Response files are XML files. You can include a directory location when you enter the response file name to save the file to a different location.
13. Enter I: Install.
14. When the installation completes, enter F: Finish.

Installing silently by using a response file

You can use a response file to install in silent mode.

Before you begin

Locate the SilentInstallOptions response file provided in the installation package. You must also shut down the application server and the IBM® SPSS® Collaboration and Deployment Services server before starting the installation.

Procedure

To install a package in silent mode:

Run the **imcl** command:

- Windows: **imcl.exe**
`input response_file -log log_file`
- Linux® and UNIX: **./imcl input response_file -log log_file**

The default **imcl** location will vary depending on the operating system and installation type (administrator, nonadministrator, or group). For more information, see [Getting started with Installation Manager](#).

Results

When the installation is complete, a status of 0 is returned. If the installation cannot be completed, a non-zero number is returned.
A log file is available. For more information, see the Installation Manager documentation.

Example

Table 1. Install commands by operating system

Operating system	Command
Windows	<code>imcl.exe input c:\response_files\install.xml -log c:\mylog\install_log.xml -acceptLicense</code>

Operating system	Command
Linux, UNIX	<code>./imcl input /response_files/install.xml -log /mylog/install_log.xml -acceptLicense</code>

Enclose file paths that include spaces with double quotation marks.

Configuring the adapter for IBM SPSS Collaboration and Deployment Services Web services on Linux

When running IBM® SPSS® Collaboration and Deployment Services Web service automations relating to SPSS Modeler on Linux, you may see the error:

```
java.io.IOException: Too many open files
```

If this occurs, increase the maximum number of open files on the Linux server by entering the command:

```
ulimit -n value
```

where *value* is the number of files that can be open. This value should be as high as possible, and depends on the amount of nodes in stream files. Default is 1024; a suggested value is 100000.

Configuring the adapter for SPSS Statistics

If you want to use the IBM® SPSS® Collaboration and Deployment Services scoring service to score an SPSS Modeler stream containing SPSS Statistics integration nodes (Statistics Transform, Statistics Model, and Statistics Output), perform the following procedure after installing the adapter.

1. On the repository host, navigate to the `/components/modeler/bin` folder.
2. Use the `statisticsutility` tool to configure the adapter to work with the SPSS Statistics server. For more information, see the section on IBM SPSS Statistics helper applications in the *IBM SPSS Modeler Source, Process and Output Nodes* guide.

Troubleshooting

IBM Installation Manager version

IBM Installation Manager version **1.9.1** is required for the installation.

Free disk space

The installation will fail if you don't have sufficient free disk space. A minimum of 10 GB is recommended.

Installation failure

IBM Installation Manager may report that installation is successful, even though the adapter is not functioning correctly. The installation consists of an *installation* process and a *configuration* process. In some cases, the installation process may succeed but configuration process may fail. Verify that the entire installation is successful before continuing.

Uninstalling IBM SPSS Modeler Server Adapter

You can uninstall IBM® SPSS® Modeler Server Adapter in wizard or console mode.

You must log in with a user account that has the same privileges as the account that was used to install IBM SPSS Modeler Server Adapter.

Important: Certain files in the IBM SPSS Modeler Server Adapter program directory (for example, program data) cannot be deleted by IBM Installation Manager. You must manually delete the program directory to completely remove all IBM SPSS Modeler Server Adapter files from the system after you uninstall it.

Uninstalling IBM SPSS Modeler Server Adapter removes the adapter files from the host file system. However, the adapter remains deployed with IBM SPSS Collaboration and Deployment Services Repository in the application server.

- [Uninstalling by using wizard mode](#)
You can use IBM Installation Manager in wizard mode to uninstall IBM SPSS Modeler Server Adapter.
- [Uninstalling by using console mode](#)
You can use IBM Installation Manager in console mode to uninstall IBM SPSS Modeler Server Adapter.

Uninstalling by using wizard mode

You can use IBM Installation Manager in wizard mode to uninstall IBM SPSS Modeler Server Adapter.

Before you begin

You must log in with a user account that has the same privileges as the account that was used to install the packages that you want to uninstall.

Procedure

To uninstall IBM SPSS Modeler Server Adapter:

1. Close programs that you installed with Installation Manager.
2. Start Installation Manager in wizard mode by using **IBMIM**.
The default **IBMIM** location will vary depending on the operating system and installation type (administrator, nonadministrator, or group). For more information, see [Getting started with Installation Manager](#).
3. In Installation Manager, click Uninstall.
4. In the Uninstall wizard, select the IBM SPSS Modeler Server Adapter package.
5. Click Next.
6. On the Summary page, review your selections. Click Back to change your selections. If you are satisfied with your choices, click Uninstall.
On Windows, Installation Manager checks for running processes. If processes are blocking the uninstall process, a list of these processes is shown in the Blocking Processes section. You must stop these processes before you continue the uninstall process. Click Stop All Blocking Processes. If there are no processes that must be stopped, you do not see this list. The running processes lock files that must be accessed or modified by Installation Manager.
7. When the uninstallation process finishes, the Complete page opens and confirms the uninstallation process.

Uninstalling by using console mode

You can use IBM Installation Manager in console mode to uninstall IBM SPSS Modeler Server Adapter.

Before you begin

You must log in with a user account that has the same privileges as the account that was used to install the packages.

About this task

A selected option is indicated by an **x** in brackets: **[x]**. Options that are not selected are indicated by empty brackets: **[]**. You can press Enter to select the default entry or select a different command. For example, **[N]** indicates that the default selection is N for the **Next** command.

Procedure

To uninstall IBM SPSS Modeler Server Adapter:

1. Close programs that you installed with Installation Manager.
2. Start Installation Manager in console mode by using **imcl -c**.
The default **imcl** location will vary depending on the operating system and installation type (administrator, nonadministrator, or group). For more information, see [Getting started with Installation Manager](#).
3. Enter 5: Uninstall - Remove the installed software packages.
4. To select the IBM SPSS Modeler Server Adapter package group, enter the number that is next to the package group.
5. Enter **N**: Next to continue.
6. To select the package, enter the number that is next to the package.
Optional: To select all packages to uninstall, enter **A**: Select all packages. The **A**: Unselect all packages option shows when all packages are selected for uninstall.

7. In the Summary panel, review your selections before you uninstall. Enter **U**: Uninstall.
 8. When the uninstall process completes, enter **F**: Finish.
-

About IBM SPSS Modeler

IBM® SPSS® Modeler is a set of data mining tools that enable you to quickly develop predictive models using business expertise and deploy them into business operations to improve decision making. Designed around the industry-standard CRISP-DM model, IBM SPSS Modeler supports the entire data mining process, from data to better business results.

IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics. The methods available on the Modeling palette allow you to derive new information from your data and to develop predictive models. Each method has certain strengths and is best suited for particular types of problems.

SPSS Modeler can be purchased as a standalone product, or used as a client in combination with SPSS Modeler Server. A number of additional options are also available, as summarized in the following sections. For more information, see
<https://www.ibm.com/analytics/us/en/technology/spss/>.

- [IBM SPSS Modeler Products](#)
 - [IBM SPSS Modeler Editions](#)
 - [Documentation](#)
 - [Application examples](#)
 - [Demos Folder](#)
 - [License tracking](#)
-

IBM SPSS Modeler Products

The IBM® SPSS® Modeler family of products and associated software comprises the following.

- IBM SPSS Modeler
 - IBM SPSS Modeler Server
 - IBM SPSS Modeler Administration Console (included with IBM SPSS Deployment Manager)
 - IBM SPSS Modeler Batch
 - IBM SPSS Modeler Solution Publisher
 - IBM SPSS Modeler Server adapters for IBM SPSS Collaboration and Deployment Services
-
- [IBM SPSS Modeler](#)
 - [IBM SPSS Modeler Server](#)
 - [IBM SPSS Modeler Administration Console](#)
 - [IBM SPSS Modeler Batch](#)
 - [IBM SPSS Modeler Solution Publisher](#)
 - [IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services](#)
-

IBM SPSS Modeler

SPSS® Modeler is a functionally complete version of the product that you install and run on your personal computer. You can run SPSS Modeler in local mode as a standalone product, or use it in distributed mode along with IBM® SPSS Modeler Server for improved performance on large data sets.

With SPSS Modeler, you can build accurate predictive models quickly and intuitively, without programming. Using the unique visual interface, you can easily visualize the data mining process. With the support of the advanced analytics embedded in the product, you can discover previously hidden patterns and trends in your data. You can model outcomes and understand the factors that influence them, enabling you to take advantage of business opportunities and mitigate risks.

SPSS Modeler is available in two editions: SPSS Modeler Professional and SPSS Modeler Premium. See the topic [IBM SPSS Modeler Editions](#) for more information.

IBM SPSS Modeler Server

SPSS® Modeler uses a client/server architecture to distribute requests for resource-intensive operations to powerful server software, resulting in faster performance on larger data sets.

SPSS Modeler Server is a separately-licensed product that runs continually in distributed analysis mode on a server host in conjunction with one or more IBM® SPSS Modeler installations. In this way, SPSS Modeler Server provides superior performance on large data sets because memory-intensive operations can be done on the server without downloading data to the client computer. IBM SPSS Modeler Server also provides support for SQL optimization and in-database modeling capabilities, delivering further benefits in performance and automation.

Related information

- [About IBM SPSS Modeler](#)
 - [Documentation](#)
 - [Overview of modeling nodes](#)
-

IBM SPSS Modeler Administration Console

The Modeler Administration Console is a graphical user interface for managing many of the SPSS® Modeler Server configuration options, which are also configurable by means of an options file. The console is included in IBM® SPSS Deployment Manager, can be used to monitor and configure your SPSS Modeler Server installations, and is available free-of-charge to current SPSS Modeler Server customers. The application can be installed only on Windows computers; however, it can administer a server installed on any supported platform.

IBM® SPSS Modeler Batch

While data mining is usually an interactive process, it is also possible to run SPSS® Modeler from a command line, without the need for the graphical user interface. For example, you might have long-running or repetitive tasks that you want to perform with no user intervention. SPSS Modeler Batch is a special version of the product that provides support for the complete analytical capabilities of SPSS Modeler without access to the regular user interface. SPSS Modeler Server is required to use SPSS Modeler Batch.

IBM SPSS Modeler Solution Publisher

SPSS® Modeler Solution Publisher is a tool that enables you to create a packaged version of an SPSS Modeler stream that can be run by an external runtime engine or embedded in an external application. In this way, you can publish and deploy complete SPSS Modeler streams for use in environments that do not have SPSS Modeler installed. SPSS Modeler Solution Publisher is distributed as part of the IBM® SPSS Collaboration and Deployment Services - Scoring service, for which a separate license is required. With this license, you receive SPSS Modeler Solution Publisher Runtime, which enables you to execute the published streams.

For more information about SPSS Modeler Solution Publisher, see the IBM SPSS Collaboration and Deployment Services documentation. The IBM SPSS Collaboration and Deployment Services IBM Documentation contains sections called "IBM SPSS Modeler Solution Publisher" and "IBM SPSS Analytics Toolkit."

IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services

A number of adapters for IBM® SPSS® Collaboration and Deployment Services are available that enable SPSS Modeler and SPSS Modeler Server to interact with an IBM SPSS Collaboration and Deployment Services repository. In this way, an SPSS Modeler stream deployed to the repository can be shared by multiple users, or accessed from the thin-client application IBM SPSS Modeler Advantage. You install the adapter on the system that hosts the repository.

IBM SPSS Modeler Editions

SPSS® Modeler is available in the following editions.

SPSS Modeler Professional

SPSS Modeler Professional provides all the tools you need to work with most types of structured data, such as behaviors and interactions tracked in CRM systems, demographics, purchasing behavior and sales data.

SPSS Modeler Premium

SPSS Modeler Premium is a separately-licensed product that extends SPSS Modeler Professional to work with specialized data and with unstructured text data. SPSS Modeler Premium includes IBM® SPSS Modeler Text Analytics:

IBM SPSS Modeler Text Analytics uses advanced linguistic technologies and Natural Language Processing (NLP) to rapidly process a large variety of unstructured text data, extract and organize the key concepts, and group these concepts into categories. Extracted concepts and categories can be combined with existing structured data, such as demographics, and applied to modeling using the full suite of IBM SPSS Modeler data mining tools to yield better and more focused decisions.

IBM SPSS Modeler Subscription

IBM SPSS Modeler Subscription provides all the same predictive analytics capabilities as the traditional IBM SPSS Modeler client. With the Subscription edition, you can download product updates regularly.

Documentation

Documentation is available from the Help menu in SPSS® Modeler. This opens the online IBM Documentation, which is always available outside the product.

Complete documentation for each product (including installation instructions) is also available in PDF format at <https://www.ibm.com/support/pages/spss-modeler-184-documentation>.

- [SPSS Modeler Professional Documentation](#)
 - [SPSS Modeler Premium Documentation](#)
-

SPSS Modeler Professional Documentation

The SPSS® Modeler Professional documentation suite (excluding installation instructions) is as follows.

- **IBM® SPSS Modeler User's Guide.** General introduction to using SPSS Modeler, including how to build data streams, handle missing values, build CLEM expressions, work with projects and reports, and package streams for deployment to IBM SPSS Collaboration and Deployment Services or IBM SPSS Modeler Advantage.
- **IBM SPSS Modeler Source, Process, and Output Nodes.** Descriptions of all the nodes used to read, process, and output data in different formats. Effectively this means all nodes other than modeling nodes.
- **IBM SPSS Modeler Modeling Nodes.** Descriptions of all the nodes used to create data mining models. IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics.
- **IBM SPSS Modeler Applications Guide.** The examples in this guide provide brief, targeted introductions to specific modeling methods and techniques. An online version of this guide is also available from the Help menu. See the topic [Application examples](#) for more information.
- **IBM SPSS Modeler Python Scripting and Automation.** Information on automating the system through Python scripting, including the properties that can be used to manipulate nodes and streams.
- **IBM SPSS Modeler Deployment Guide.** Information on running IBM SPSS Modeler streams as steps in processing jobs under IBM SPSS Deployment Manager.
- **IBM SPSS Modeler In-Database Mining Guide.** Information on how to use the power of your database to improve performance and extend the range of analytical capabilities through third-party algorithms.
- **IBM SPSS Modeler Server Administration and Performance Guide.** Information on how to configure and administer IBM SPSS Modeler Server.
- **IBM SPSS Deployment Manager User Guide.** Information on using the administration console user interface included in the Deployment Manager application for monitoring and configuring IBM SPSS Modeler Server.
- **IBM SPSS Modeler CRISP-DM Guide.** Step-by-step guide to using the CRISP-DM methodology for data mining with SPSS Modeler.
- **IBM SPSS Modeler Batch User's Guide.** Complete guide to using IBM SPSS Modeler in batch mode, including details of batch mode execution and command-line arguments. This guide is available in PDF format only.

SPSS Modeler Premium Documentation

The SPSS® Modeler Premium documentation suite (excluding installation instructions) is as follows.

- **SPSS Modeler Text Analytics User's Guide.** Information on using text analytics with SPSS Modeler, covering the text mining nodes, interactive workbench, templates, and other resources.

Application examples

While the data mining tools in SPSS® Modeler can help solve a wide variety of business and organizational problems, the application examples provide brief, targeted introductions to specific modeling methods and techniques. The data sets used here are much smaller than the enormous data stores managed by some data miners, but the concepts and methods that are involved are scalable to real-world applications.

To access the examples, click Application Examples on the Help menu in SPSS Modeler.

The data files and sample streams are installed in the Demos folder under the product installation directory. For more information, see [Demos Folder](#).

A PDF version of the Applications Guide is also available. For more information, see [Documentation](#).

Demos Folder

The data files and sample streams that are used with the application examples are installed in the Demos folder under the product installation directory (for example: C:\Program Files\IBM\SPSS\Modeler\<version>\Demos). This folder can also be accessed from the IBM SPSS® Modeler program group on the Windows Start menu, or by clicking Demos on the list of recent directories in the File...Open Stream dialog box.

License tracking

When you use SPSS® Modeler, license usage is tracked and logged at regular intervals. The license metrics that are logged are *AUTHORIZED_USER* and *CONCURRENT_USER*, and the type of metric that is logged depends on the type of license that you have for SPSS Modeler.

The log files that are produced can be processed by the IBM License Metric Tool, from which you can generate license usage reports.

The license log files are created in the same directory where SPSS Modeler Client log files are recorded (by default, %ALLUSERSPROFILE%/IBM/SPSS/Modeler/<version>/log).

What's new in version 18.4.0?

IBM® SPSS® Modeler adds the following features and enhancements in this release.

- Text Analytics flows created in Cloud Pak for Data can now be exported from Cloud Pak for Data (in JSON template format) and imported to a IBM SPSS Modeler stream.
- You can now switch Python environments via the SPSS Modeler user interface. For details, see [Switching to a different Python environment](#).
- SPSS Modeler now provides enhanced password encryption. Due to a backward compatibility issue, note the following:
 - If you run SPSS Modeler Server using a private password file to authenticate users against a private password database, you must use the **pwutil** executable to recreate the database in 18.4. For instructions, see [Configuring a private password database](#).
 - If you use an encrypted password (via the Encode Password option on the Tools menu), you must recreate it in version 18.4. For more information, see [Generating an encoded password](#).
- A new menu item is available for managing your SPSS Modeler Server connections: Tools...Manage Modeler Server Configuration. The available settings are similar to those found in the SPSS Modeler Server Configuration section of the Modeler Administration Console included with SPSS Collaboration and Deployment Services Deployment Manager. For details about the settings, see [SPSS Modeler Server Configuration](#).
- Empty passwords are no longer supported for Cognos TM1 Server connections.
- Windows 11 is now supported.
- MacOS Monterey (also known as MacOS 12) is now supported.
- Java 11 is now supported.
- CPLEX 22.1 is now supported.
- InstallAnywhere 2021 sp2 is now supported.
- SPSS Data Access Pack 8.1.1 is now supported
- Netezza Performance Server 11.x is now supported.
- Cognos Analytics Connector 11.1.7 is now supported.

- Amazon S3 is now supported through Athena (for reading data, only).
 - ClickHouse database version 22.3 is now supported with the following limitations:
 - SQL pushback isn't supported for the Merge function in the database export node (due to weak support by ClickHouse for update, delete, etc.)
 - SQL pushback isn't supported for the *time* type (due to there being no time type in ClickHouse)
 - The *date* baseline is 1970-01-01
-

Lists and geospatial data

The ability to process lists and geospatial data was added in IBM® SPSS® Modeler 17. This topic lists some of the areas where you can find further information about lists and geospatial data.

Lists

- For information about the list storage type and list depths, see [List storage and associated measurement levels](#).
- For information about creating a list from the Derive node, see [Deriving a list or geospatial field](#).
- For information about creating a list from the Merge node by merging geospatial data, see [Specifying Ranked Conditions for a Merge](#).
- For information about list values, see [Setting derived list values](#).
- For information about list icons, see [Setting Field Storage and Formatting](#).
- For information about the collection measurement level, see [Specifying Values for Collection Data](#).
- For information about one of the modeling nodes that can process non-geospatial list data, see [Association Rules node](#).

Geospatial data

- For information about importing geospatial data from a `.shp` file or a map service, see [Geospatial Source Node](#).
- For information about importing geospatial data as a flat file, see [Importing geospatial data into the Variable File Node](#).
- For information about creating a geospatial field from the Derive node, see [Deriving a list or geospatial field](#).
- For information about setting a default coordinate system for geospatial data, see [Setting geospatial options for streams](#).
- For information about changing the coordinate system for imported geospatial data, see [Reprojection Node](#).
- For information about the geospatial measurement level, see [Specifying values for geospatial data](#).
- For information about the geospatial measurement sublevels and associated restrictions, see [Geospatial measurement sublevels](#).
- For information about merging geospatial data, see [Specifying Ranked Conditions for a Merge](#).
- For an example of list depths in geospatial data, see [List storage and associated measurement levels](#).
- For a list of the geospatial functions that you can use in the Expression Builder, see [Spatial functions](#).
- For information about one of the modeling nodes that can process geospatial data, see [Spatio-Temporal Prediction modeling node](#).

Product overview

- [Getting started](#)
- [Starting IBM SPSS Modeler](#)
- [IBM SPSS Modeler Interface at a Glance](#)
- [Printing](#)
- [Automating IBM SPSS Modeler](#)

Getting started

Welcome to IBM® SPSS® Modeler, a member of the modeling family of products from IBM Corp. In case you're wondering where to start, here are some suggestions.

You can...

- Try [building a stream](#).
- See [what's new in this release](#).
- Visit the IBM SPSS Modeler section of the IBM Analytics web site at <https://www.ibm.com/analytics/us/en/technology/spss/>.
- Visit the corporate Support site at <http://www.ibm.com/support>, choosing Information Management as the support type.

There's also an interactive tutorial for IBM SPSS Modeler. On the main menu, click:

Help>Application Examples

You can also learn about...

- [Data mining and modeling](#)
- [Predictive analytics](#)
- [Modeling techniques](#)
- [Which technique to use](#)

About IBM SPSS Modeler

- [How it can help you](#)
 - [Where you can use it](#)
 - [How it works](#)
 - [How to use it](#)
 - [Data Mining and Modeling](#)
 - [About Predictive Analytics](#)
 - [Modeling Techniques](#)
 - [How Do I Know Which Technique To Use?](#)
 - [How IBM SPSS Modeler Can Help You](#)
 - [Where You Can Use IBM SPSS Modeler](#)
 - [How IBM SPSS Modeler works](#)
 - [How to Use IBM SPSS Modeler](#)
-

Data Mining and Modeling

Data mining is the process of digging down into your business data to discover hidden patterns and relationships. Data mining solves a common problem: the more data you have, the more difficult and time-consuming it is to analyze and draw meaning from the data effectively. What should be a gold mine often lies unexplored, owing to a lack of personnel, time, or expertise.

Data mining uses a clear business procedure and powerful analytic technologies to quickly and thoroughly explore mountains of data, pulling out and presenting you with the valuable, usable information - the business intelligence - that you need.

While those previously unknown patterns and relationships in your data are interesting in themselves, that's not the end of the story. What if you could use those patterns of past behavior to predict what might happen in the future? That's where modeling comes in - a **model** is the set of rules, formulas or equations extracted from your source data and enabling you to generate predictions against that. This is the heart of predictive analytics.

About Predictive Analytics

Predictive analytics is a business process and a set of related technologies that helps to link up your data with effective action by drawing reliable conclusions about current conditions and future events. It is a combination of:

- Advanced analytics
- Decision optimization

Advanced analytics uses a number of tools and techniques to analyze past and present events and predict future outcomes. **Decision optimization** determines which actions on your part will produce the best possible outcomes, ensuring that those recommended actions are delivered to where they can most effectively be incorporated into your business processes.

Modeling Techniques

Modeling techniques are based around the use of algorithms - sequences of instructions for solving specific problems. You use a particular algorithm to create that type of model. There are three main classes of modeling technique, and IBM® SPSS® Modeler provides several examples of each:

- Supervised
- Association
- Segmentation (sometimes known as "clustering")

Supervised models use the values of one or more **input** fields to predict the value of one or more output, or **target**, fields. Some examples of these techniques are: decision trees (C&R Tree, QUEST, CHAID and C5.0 algorithms), regression (linear, logistic, generalized linear, and Cox

regression algorithms), neural networks, support vector machines, and Bayesian networks.

Association models find patterns in your data where one or more entities (such as events, purchases, or attributes) are associated with one or more other entities. The models construct rule sets that define these relationships. Here the fields within the data can act as both inputs and targets. You could find these associations manually, but association rule algorithms do so much more quickly, and can explore more complex patterns. Apriori and Carma models are examples of the use of such algorithms. One other type of association model is a sequence detection model, which finds sequential patterns in time-structured data.

Segmentation models divide the data into segments, or clusters, of records that have similar patterns of input fields. As they are only interested in the input fields, segmentation models have no concept of output or target fields. Examples of segmentation models are Kohonen networks, K-Means clustering, two-step clustering and anomaly detection.

How Do I Know Which Technique To Use?

With such a wide variety of techniques at your disposal, it can be difficult to know where to start to solve your particular problem. Fortunately, IBM® SPSS® Modeler can make some of these decisions for you, in the form of **automated modeling**. This is a very powerful technique that estimates and compares a number of different modeling methods, ranking them in order of effectiveness. In this way you can try out a variety of approaches in a single modeling run.

As you become more familiar with the various techniques, you will learn which algorithms, and which options within those algorithms, are the most suitable for your particular problems.

How IBM SPSS Modeler Can Help You

IBM® SPSS® Modeler is a data mining, modeling and reporting tool. It gives you a unique graphical interface that enables you quickly and easily to uncover patterns and trends in your data. You can then use the statistical algorithms built into the product's advanced analytics to create data models that predict future outcomes from your current and historical data.

Where You Can Use IBM SPSS Modeler

You can use IBM® SPSS® Modeler to solve a huge variety of business problems. If you have a problem that involves data, the chances are that IBM SPSS Modeler can provide a solution. Here are just a few examples of industries and applications where IBM SPSS Modeler can make a difference.

Education. A community college wants to gain a deep understanding of student enrollment patterns and tendencies in order to improve student retention. By predicting which students are less likely to return to school, faculty and management can directly or indirectly intervene with academic counseling, financial aid packages, and curriculum offerings.

Financial. By analyzing its customer information, a mortgage lender estimates which customers are likely to move to another provider at the end of their discounted or fixed-rate period, which customers will switch to a new rate, and which ones will revert to the standard rate.

Crime detection. A local law enforcement agency analyzes a file of crime reports looking for patterns in the data. They segment the data to find clusters of crimes with similar times, locations and methods, and which therefore may have been committed by the same offender.

Non-profit organization. Taking as input a list of past donors and the amount of their donations, a charity calculates the probability of a positive response to a direct mail campaign. Using the list of likely positive responders, the charity then predicts the donation amount, calculates the probably return on the campaign, and constructs a mailing list of potential donors.

Manufacturing. An industrial manufacturing company monitors the operating condition of various pieces of machinery in order to predict possible breakdowns before they occur.

Medical. A pharmaceutical company wants to know which of several possible treatments is likely to be the most effective for a set of patients who suffer from the same disease. By analyzing which drugs show a better response rate for particular symptoms, the company builds a model that predicts which medication is best suited for a particular type of patient.

Retail. By analyzing electronic point-of-sale data from its tills, a supermarket can discover which customers with loyalty cards typically buy which combinations of products. Using the results of this market basket analysis, the retailer can target customers more accurately for promotions, improve response rates from mailshots, or adjust stock ranges for specific types of demographic populations in different branches.

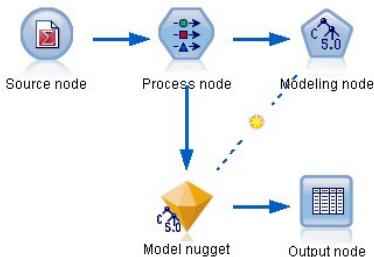
Telecommunications. As part of its efforts to reduce customer churn (loss of business), a telecommunications company is interested in modeling the "time to churn" in order to determine the factors that are associated with customers who are quick to switch to another service. To this end, a random sample of customers is selected and various data about them are pulled from the database and analyzed. The company then

models the time to churn, plotting the expected number of customers retained over the next two years, and identifying the individual customers most likely to churn in the next year.

Some of these stories, together with a number of others, are available as fully worked examples in the *IBM SPSS Modeler Applications Guide*.

How IBM SPSS Modeler works

Figure 1. Nodes in a stream



The unique graphical interface in IBM® SPSS® Modeler is based around nodes and streams. Nodes are the icons or shapes that represent individual operations on your data. The nodes are linked together in a stream to represent the flow of data through each operation.

Algorithms are represented by a special type of node known as a modeling node. There is a different modeling node for each algorithm that IBM SPSS Modeler supplies. Modeling nodes are shown as a five-sided shape.

Other types of nodes include source nodes, process nodes, and output nodes. Source nodes are the ones that bring the data into the stream, and always appear at the beginning of the stream. Process nodes perform operations on individual data records and fields, and are usually found in the middle of the stream. Output nodes produce a variety of output for data, charts and model results, or they enable you to export the results to another application, such as a database or a spreadsheet. Output nodes usually appear as the last node in a stream or a branch of a stream.

When you run a stream that contains a modeling node, the resulting model is added to stream, and is represented by a special type of node known as a model nugget--it has a shape that looks just like a gold nugget.

How to Use IBM SPSS Modeler

You can use IBM® SPSS® Modeler for ad-hoc analysis on your data to discover previous hidden patterns or trends. However, to get real value from this powerful piece of software, you need a more careful, methodical approach.

A simple, structured approach might go something like this:

- Understand your business and what you are trying to achieve with data mining
- Examine and understand your data
- Prepare the data for analysis and modeling
- Build predictive models
- Evaluate the models to verify that they are correct
- Deploy the models into your organization

This sequence is actually the basis for a more in-depth structured approach to data mining known as CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining. IBM SPSS Modeler is specifically designed to work hand-in-hand with this approach.

For more information, on the main menu click:

Help > CRISP-DM Help

Starting IBM® SPSS® Modeler

To start the application, click:

Start > [All] Programs > IBM SPSS Modeler <version> > IBM SPSS Modeler <version>

The main window is displayed after a few seconds.

- [Launching from the Command Line](#)

- [Connecting to IBM SPSS Modeler Server](#)
- [Connecting to Analytic Server](#)
- [Changing the temp directory](#)
- [Starting Multiple IBM SPSS Modeler Sessions](#)

Launching from the Command Line

You can use the command line of your operating system to launch IBM® SPSS® Modeler as follows:

1. On a computer where IBM SPSS Modeler is installed, open a DOS, or command-prompt, window.
2. To launch the IBM SPSS Modeler interface in interactive mode, type the `modelerclient` command followed by the required arguments; for example:

```
modelerclient -stream report.str -execute
```

The available arguments (flags) allow you to connect to a server, load streams, run scripts, or specify other parameters as needed.

Connecting to IBM SPSS Modeler Server

IBM® SPSS® Modeler can be run as a standalone application, or as a client connected to IBM SPSS Modeler Server directly or to an IBM SPSS Modeler Server or server cluster through the Coordinator of Processes plug-in from IBM SPSS Collaboration and Deployment Services. The current connection status is displayed at the bottom left of the IBM SPSS Modeler window.

Whenever you want to connect to a server, you can manually enter the server name to which you want to connect or select a name that you have previously defined. However, if you have IBM SPSS Collaboration and Deployment Services, you can search through a list of servers or server clusters from the Server Login dialog box. The ability to browse through the Statistics services running on a network is made available through the Coordinator of Processes.

To Connect to a Server

1. On the Tools menu, click Server Login. The Server Login dialog box opens. Alternatively, double-click the connection status area of the IBM SPSS Modeler window.
2. Using the dialog box, specify options to connect to the local server computer or select a connection from the table.
 - Click Add or Edit to add or edit a connection. See the topic [Adding and Editing the IBM SPSS Modeler Server Connection](#) for more information.
 - Click Search to access a server or server cluster in the Coordinator of Processes. See the topic [Searching for Servers in IBM SPSS Collaboration and Deployment Services](#) for more information.

Server table. This table contains the set of defined server connections. The table displays the default connection, server name, description, and port number. You can manually add a new connection, as well as select or search for an existing connection. To set a particular server as the default connection, select the check box in the Default column in the table for the connection.

Default data path. Specify a path used for data on the server computer. Click the ellipsis button (...) to browse to the required location.

Set Credentials. Leave this box unchecked to enable the **single sign-on** feature, which attempts to log you in to the server using your local computer username and password details. If single sign-on is not possible, or if you check this box to disable single sign-on (for example, to log in to an administrator account), the following fields are enabled for you to enter your credentials.

User ID. Enter the user name with which to log on to the server.

Password. Enter the password associated with the specified user name.

Domain. Specify the domain used to log on to the server. A domain name is required only when the server computer is in a different Windows domain than the client computer.

3. Click OK to complete the connection.

To Disconnect from a Server

1. On the Tools menu, click Server Login. The Server Login dialog box opens. Alternatively, double-click the connection status area of the IBM SPSS Modeler window.
2. In the dialog box, select the Local Server and click OK.
 - [Adding and Editing the IBM SPSS Modeler Server Connection](#)
 - [Searching for Servers in IBM SPSS Collaboration and Deployment Services](#)
 - [Configuring single sign-on](#)
 - [Adding and Editing the IBM SPSS Modeler Server Connection](#)
 - [Searching for Servers in IBM SPSS Collaboration and Deployment Services](#)

Related information

- [Starting IBM SPSS Modeler](#)
-

Adding and Editing the IBM SPSS Modeler Server Connection

You can manually edit or add a server connection in the Server Login dialog box. By clicking Add, you can access an empty Add/Edit Server dialog box in which you can enter server connection details. By selecting an existing connection and clicking Edit in the Server Login dialog box, the Add/Edit Server dialog box opens with the details for that connection so that you can make any changes.

Note: You cannot edit a server connection that was added from IBM® SPSS® Collaboration and Deployment Services, since the name, port, and other details are defined in IBM SPSS Collaboration and Deployment Services. Best practice dictates that the same ports should be used to communicate with both IBM SPSS Collaboration and Deployment Services and SPSS Modeler Client. These can be set as `max_server_port` and `min_server_port` in the options.cfg file.

To Add Server Connections

1. On the Tools menu, click Server Login. The Server Login dialog box opens.
 2. In this dialog box, click Add. The Server Login Add/Edit Server dialog box opens.
 3. Enter the server connection details and click OK to save the connection and return to the Server Login dialog box.
- **Server.** Specify an available server or select one from the list. The server computer can be identified by an alphanumeric name (for example, *myserver*) or an IP address assigned to the server computer (for example, 202.123.456.78).
 - **Port.** Give the port number on which the server is listening. If the default does not work, ask your system administrator for the correct port number.
 - **Description.** Enter an optional description for this server connection.
 - **Ensure secure connection (use SSL).** Specifies whether an SSL (**Secure Sockets Layer**) connection should be used. SSL is a commonly used protocol for securing data sent over a network. To use this feature, SSL must be enabled on the server hosting IBM SPSS Modeler Server. If necessary, contact your local administrator for details.

To Edit Server Connections

1. On the Tools menu, click Server Login. The Server Login dialog box opens.
2. In this dialog box, select the connection you want to edit and then click Edit. The Server Login Add/Edit Server dialog box opens.
3. Change the server connection details and click OK to save the changes and return to the Server Login dialog box.

Related information

- [Starting IBM SPSS Modeler](#)
-

Searching for Servers in IBM SPSS Collaboration and Deployment Services

Instead of entering a server connection manually, you can select a server or server cluster available on the network through the Coordinator of Processes, available in IBM® SPSS® Collaboration and Deployment Services. A server cluster is a group of servers from which the Coordinator of Processes determines the server best suited to respond to a processing request.

Although you can manually add servers in the Server Login dialog box, searching for available servers lets you connect to servers without requiring that you know the correct server name and port number. This information is automatically provided. However, you still need the correct logon information, such as username, domain, and password.

Note: If you do not have access to the Coordinator of Processes capability, you can still manually enter the server name to which you want to connect or select a name that you have previously defined. See the topic [Adding and Editing the IBM SPSS Modeler Server Connection](#) for more information.

To search for servers and clusters

1. On the Tools menu, click Server Login. The Server Login dialog box opens.
2. In this dialog box, click Search to open the Search for Servers dialog box. If you are not logged on to IBM SPSS Collaboration and Deployment Services when you attempt to browse the Coordinator of Processes, you will be prompted to do so.
3. Select the server or server cluster from the list.
4. Click OK to close the dialog box and add this connection to the table in the Server Login dialog box.

Related information

- [Starting IBM SPSS Modeler](#)
-

Connecting to Analytic Server

If you have multiple Analytic Servers available, you can use the Analytic Server Connection dialog to define more than one server for use in IBM® SPSS® Modeler. Your administrator may have already set up a default Analytic Server in the <Modeler_install_path>/config/options.cfg file. But you can also use other available servers after defining them. For example, when using the Analytic Server Source and Export nodes, you may want to use different Analytic Server connections in different branches of a stream so that when each branch runs it uses its own Analytic Server and no data will be pulled to the IBM SPSS Modeler Server. Note that if a branch contains more than one Analytic Server connection, the data will be pulled from the Analytic Servers to the IBM SPSS Modeler Server.

To create a new Analytic Server connection, go to Tools > Analytic Server Connections and provide the required information in the following sections of the dialog.

Connection

URL. Type the URL for the Analytic Server in the format `https://hostname:port/contextroot`, where `hostname` is the IP address or host name of the Analytic Server, `port` is its port number, and `contextroot` is the context root of the Analytic Server.

Tenant. Type the name of the tenant that the IBM SPSS Modeler Server is a member of. Contact your administrator if you don't know the tenant.

Authentication

Mode. Select from the following authentication modes.

- Username and password requires you to enter the username and password.
- Stored credential requires you to select a credential from the IBM SPSS Collaboration and Deployment Services Repository.
- Kerberos requires you to enter the service principal name and the config file path. Contact your administrator if you don't know this information.

Username. Type the Analytic Server username.

Reams. Select the realm to use for the Analytic Server connection.

Password. Type the Analytic Server password.

Connect. Click Connect to test the new connection.

Connections

After specifying the information above and clicking Connect, the connection will be added to this Connections table. If you need to remove a connection, select it and click Remove.

If your administrator defined a default Analytic Server connection in the options.cfg file, you can click Add default connection to add it to your available connections also. You will be prompted for the username and password.

For information about setting up a default connection in options.cfg, see [Options visible in options.cfg](#).

Changing the temp directory

Some operations that are done by IBM® SPSS® Modeler Server might require temporary files to be created. By default, IBM SPSS Modeler uses the system temporary directory to create temp files. You can alter the location of the temporary directory using the following steps.

1. Create a new directory called spss and subdirectory called servertemp.
2. Edit options.cfg, located in the /config directory of your IBM SPSS Modeler installation directory. Edit the `temp_directory` parameter in this file to read: `temp_directory, "C:/spss/servertemp"`.
3. Restart the IBM SPSS Modeler Server service. You can do this by clicking the Services tab on your Windows Control Panel. Stop the service and then start it to activate the changes you made. Restarting the machine also restarts the service.

All temp files now get written to this new directory.

Note: Forward slashes must be used.

Temp directory for data view

For data view service, complete the following steps to configure the temporary directory.

1. Create the temp directory D:/SPSSTemp.
2. The data view temp files are controlled by `-Djava.io.tmpdir=D:/SPSSTemp`. Add this option to the start script for data view in the following file: \${ModelerInstallDir}/dataview/start_graph_micro_service.sh.
3. Start the IBM SPSS Modeler.

Files now get written to D:/SPSSTemp.

Starting Multiple IBM SPSS Modeler Sessions

If you need to launch more than one IBM® SPSS® Modeler session at a time, you must make some changes to your IBM SPSS Modeler and Windows settings. For example, you may need to do this if you have two separate server licenses and want to run two streams against two different servers from the same client machine.

To enable multiple IBM SPSS Modeler sessions:

1. Click:
`Start > [All] Programs > IBM SPSS Modeler`
2. On the IBM SPSS Modeler shortcut (the one with the icon), right-click and select Properties.
3. In the Target text box, add `-noshare` to the end of the string.
4. In Windows Explorer, select:
`Tools > Folder Options...`
5. On the File Types tab, select the IBM SPSS Modeler Stream option and click Advanced.
6. In the Edit File Type dialog box, select Open with IBM SPSS Modeler and click Edit.
7. In the Application used to perform action text box, add `-noshare` before the `-stream` argument.

IBM SPSS Modeler Interface at a Glance

At each point in the data mining process, the easy-to-use IBM® SPSS® Modeler interface invites your specific business expertise. Modeling algorithms, such as prediction, classification, segmentation, and association detection, ensure powerful and accurate models. Model results can easily be deployed and read into databases, IBM SPSS Statistics, and a wide variety of other applications.

Working with IBM SPSS Modeler is a three-step process of working with data.

- First, you read data into IBM SPSS Modeler.
- Next, you run the data through a series of manipulations.
- Finally, you send the data to a destination.

This sequence of operations is known as a **data stream** because the data flows record by record from the source through each manipulation and, finally, to the destination--either a model or type of data output.

- [IBM SPSS Modeler Stream Canvas](#)
- [Nodes palette](#)
- [IBM SPSS Modeler Managers](#)
- [IBM SPSS Modeler Projects](#)
- [IBM SPSS Modeler Toolbar](#)
- [Customizing the Toolbar](#)
- [Customizing the IBM SPSS Modeler window](#)
- [Changing the icon size for a stream](#)
- [Using the Mouse in IBM SPSS Modeler](#)
- [Using shortcut keys](#)

Related information

- [Using the Mouse in IBM SPSS Modeler](#)
- [Using shortcut keys](#)
- [Setting IBM SPSS Modeler options](#)

IBM SPSS Modeler Stream Canvas

The stream canvas is the largest area of the IBM® SPSS® Modeler window and is where you will build and manipulate data streams.

Streams are created by drawing diagrams of data operations relevant to your business on the main canvas in the interface. Each operation is represented by an icon or **node**, and the nodes are linked together in a **stream** representing the flow of data through each operation.

You can work with multiple streams at one time in IBM SPSS Modeler, either in the same stream canvas or by opening a new stream canvas. During a session, streams are stored in the Streams manager, at the upper right of the IBM SPSS Modeler window.

Note: If using a MacBook with the built-in trackpad's Force Click and haptic feedback setting enabled, dragging and dropping nodes from the nodes palette to the stream canvas can result in duplicate nodes being added to the canvas. To avoid this issue, we recommend disabling the Force Click and haptic feedback trackpad system preference.

Nodes palette

Most of the data and modeling tools in SPSS® Modeler are available from the *Nodes Palette*, across the bottom of the window below the stream canvas.

For example, the Record Ops palette tab contains nodes that you can use to perform operations on the data *records*, such as selecting, merging, and appending.

To add nodes to the canvas, double-click icons from the Nodes Palette or drag them onto the canvas. You then connect them to create a *stream*, representing the flow of data.

Figure 1. Record Ops tab on the nodes palette



Each palette tab contains a collection of related nodes used for different phases of stream operations, such as:

- Sources nodes bring data into SPSS Modeler.
- Record Ops nodes perform operations on data *records*, such as selecting, merging, and appending.
- Field Ops nodes perform operations on data *fields*, such as filtering, deriving new fields, and determining the measurement level for given fields.
- Graphs nodes graphically display data before and after modeling. Graphs include plots, histograms, web nodes, and evaluation charts.
- Modeling nodes use the modeling algorithms available in SPSS Modeler, such as neural nets, decision trees, clustering algorithms, and data sequencing.
- Database Modeling nodes use the modeling algorithms available in Microsoft SQL Server, IBM Db2, and Oracle and Netezza databases.
- Output nodes produce various output for data, charts, and model results that can be viewed in SPSS Modeler.
- Export nodes produce various output that can be viewed in external applications, such as IBM® SPSS Data Collection or Excel.
- **IBM SPSS Statistics** nodes import data from, or export data to, IBM SPSS Statistics, as well as running IBM SPSS Statistics procedures.
- Python nodes can be used to run Python algorithms.
- Spark nodes can be used to run Spark algorithms.

As you become more familiar with SPSS Modeler, you can customize the palette contents for your own use. For more information, see [Customizing the Nodes Palette](#).

On the left side of the Nodes Palette, you can filter the nodes that display by selecting Supervised, Association, or Segmentation. For more information, see [Overview of modeling nodes](#).

Located below the Nodes Palette, a report pane provides feedback on the progress of various operations, such as when data is being read into the data stream. Also located below the Nodes Palette, a status pane provides information on what the application is currently doing, as well as indications of when user feedback is required.

Note: If using a MacBook with the built-in trackpad's Force Click and haptic feedback setting enabled, dragging and dropping nodes from the nodes palette to the stream canvas can result in duplicate nodes being added to the canvas. To avoid this issue, we recommend disabling the Force Click and haptic feedback trackpad system preference.

IBM SPSS Modeler Managers

At the top right of the window is the managers pane. This has three tabs, which are used to manage streams, output and models.

You can use the Streams tab to open, rename, save, and delete the streams created in a session.

The Outputs tab contains a variety of files, such as graphs and tables, produced by stream operations in IBM® SPSS® Modeler. You can display, save, rename, and close the tables, graphs, and reports listed on this tab.

The Models tab is the most powerful of the manager tabs. This tab contains all model **nuggets**, which contain the models generated in IBM SPSS Modeler, for the current session. These models can be browsed directly from the Models tab or added to the stream in the canvas.

IBM SPSS Modeler Projects

On the lower right side of the window is the project pane, used to create and manage data mining **projects** (groups of files related to a data mining task). There are two ways to view projects you create in IBM® SPSS® Modeler—in the Classes view and the CRISP-DM view.

The CRISP-DM tab provides a way to organize projects according to the Cross-Industry Standard Process for Data Mining, an industry-proven, nonproprietary methodology. For both experienced and first-time data miners, using the CRISP-DM tool will help you to better organize and communicate your efforts.

The Classes tab provides a way to organize your work in IBM SPSS Modeler categorically—by the types of objects you create. This view is useful when taking inventory of data, streams, and models.

IBM SPSS Modeler Toolbar

At the top of the IBM® SPSS® Modeler window, you will find a toolbar of icons that provides a number of useful functions. Following are the toolbar buttons and their functions.

	Create new stream		Open stream
	Save stream		Print current stream
	Cut & move to clipboard		Copy to clipboard
	Paste selection		Undo last action
	Redo		Search for nodes
	Edit stream properties		Preview SQL generation
	Run current stream		Run stream selection
	Stop stream (Active only while stream is running)		Add SuperNode
	Zoom in (SuperNodes only)		Zoom out (SuperNodes only)
	No markup in stream		Insert comment
	Hide stream markup (if any)		Show hidden stream markup
	Open stream in IBM SPSS Modeler Advantage		

Stream markup consists of stream comments, model links, and scoring branch indications.

Related information

- [Setting general options for streams](#)

Customizing the Toolbar

You can change various aspects of the toolbar, such as:

- Whether it is displayed
- Whether the icons have tooltips available
- Whether it uses large or small icons

To turn the toolbar display on and off:

1. On the main menu, click:
View > Toolbar > Display

To change the tooltip or icon size settings:

1. On the main menu, click:
View > Toolbar > Customize

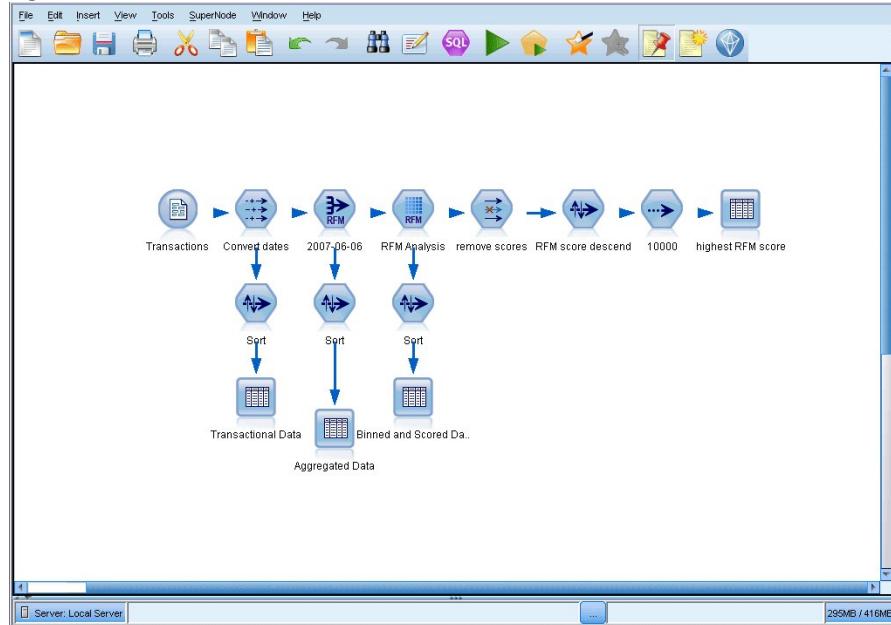
Click Show ToolTips or Large Buttons as required.

Customizing the IBM® SPSS Modeler window

Using the dividers between various portions of the SPSS® Modeler interface, you can resize or close tools to meet your preferences. For example, if you are working with a large stream, you can use the small arrows located on each divider to close the nodes palette, managers pane, and project pane. This maximizes the stream canvas, providing enough work space for large or multiple streams.

Alternatively, on the View menu, click Nodes Palette, Managers, or Project to turn the display of these items on or off.

Figure 1. Maximized stream canvas



As an alternative to closing the nodes palette, and the managers and project panes, you can use the stream canvas as a scrollable page by moving vertically and horizontally with the scrollbars at the side and bottom of the SPSS Modeler window.

You can also control the display of screen markup, which consists of stream comments, model links, and scoring branch indications. To turn this display on or off, click:

View > Stream Markup

Changing the icon size for a stream

You can change the size of the stream icons in the following ways.

- Through a stream property setting
- Through a pop-up menu in the stream
- Using the keyboard

You can scale the entire stream view to one of a number of sizes between 8% and 200% of the standard icon size.

To scale the entire stream (stream properties method)

1. From the main menu, choose Tools > Stream Properties > Options > Layout.
2. Choose the size you want from the Icon Size menu.
3. Click Apply to see the result.
4. Click OK to save the change.

To scale the entire stream (menu method)

1. Right-click the stream background on the canvas.
2. Choose Icon Size and select the size you want.

To scale the entire stream (keyboard method)

1. Press Ctrl + [-] on the main keyboard to zoom out to the next smaller size.
 2. Press Ctrl + Shift + [+] on the main keyboard to zoom in to the next larger size.
- Note that this method of zooming in may not work depending on your operating system and keyboard used.

This feature is particularly useful for gaining an overall view of a complex stream. You can also use it to minimize the number of pages needed to print a stream.

Using the Mouse in IBM SPSS Modeler

The most common uses of the mouse in IBM® SPSS® Modeler include the following:

- **Single-click.** Use either the right or left mouse button to select options from menus, open pop-up menus, and access various other standard controls and options. Click and hold the button to move and drag nodes.
- **Double-click.** Double-click using the left mouse button to place nodes on the stream canvas and edit existing nodes.
- **Middle-click.** Click the middle mouse button and drag the cursor to connect nodes on the stream canvas. Double-click the middle mouse button to disconnect a node. If you do not have a three-button mouse, you can simulate this feature by pressing the Alt key while clicking and dragging the mouse.

Related information

- [IBM SPSS Modeler Interface at a Glance](#)
- [Using shortcut keys](#)
- [Setting IBM SPSS Modeler options](#)

Using shortcut keys

Many visual programming operations in IBM® SPSS® Modeler have shortcut keys associated with them. For example, you can delete a node by clicking the node and pressing the Delete key on your keyboard. Likewise, you can quickly save a stream by pressing the S key while holding down the Ctrl key. Control commands like this one are indicated by a combination of Ctrl and another key--for example, Ctrl+S.

There are a number of shortcut keys used in standard Windows operations, such as Ctrl+X to cut. These shortcuts are supported in IBM SPSS Modeler along with the following application-specific shortcuts.

Note: In some cases, old shortcut keys used in IBM SPSS Modeler conflict with standard Windows shortcut keys. These old shortcuts are supported with the addition of the Alt key. For example, Ctrl+Alt+C can be used to toggle the cache on and off.

Table 1. Supported shortcut keys

Shortcut Key	Function
Ctrl+A	Select all
Ctrl+X	Cut

Shortcut Key	Function
Ctrl+N	New stream
Ctrl+O	Open stream
Ctrl+P	Print
Ctrl+C	Copy
Ctrl+V	Paste
Ctrl+Z	Undo
Ctrl+Q	Select all nodes downstream of the selected node
Ctrl+W	Deselect all downstream nodes (toggles with Ctrl+Q)
Ctrl+E	Run from selected node
Ctrl+S	Save current stream
Alt+Arrow keys	Move selected nodes on the stream canvas in the direction of the arrow used
Shift+F10	Open the pop-up menu for the selected node

Table 2. Supported shortcuts for old hot keys

Shortcut Key	Function
Ctrl+Alt+D	Duplicate node
Ctrl+Alt+L	Load node
Ctrl+Alt+R	Rename node
Ctrl+Alt+U	Create User Input node
Ctrl+Alt+C	Toggle cache on/off
Ctrl+Alt+F	Flush cache
Ctrl+Alt+X	Expand SuperNode
Ctrl+Alt+Z	Zoom in/zoom out
Delete	Delete node or connection

Related information

- [IBM SPSS Modeler Interface at a Glance](#)
- [Using the Mouse in IBM SPSS Modeler](#)
- [Setting IBM SPSS Modeler options](#)

Printing

The following objects can be printed in IBM® SPSS® Modeler:

- Stream diagrams
- Graphs
- Tables
- Reports (from the Report node and Project Reports)
- Scripts (from the stream properties, Standalone Script, or SuperNode script dialog boxes)
- Models (Model browsers, dialog box tabs with current focus, tree viewers)
- Annotations (using the Annotations tab for output)

To print an object:

- To print without previewing, click the Print button on the toolbar.
- To set up the page before printing, select Page Setup from the File menu.
- To preview before printing, select Print Preview from the File menu.
- To view the standard print dialog box with options for selecting printers, and specifying appearance options, select Print from the File menu.

Related information

- [IBM SPSS Modeler Toolbar](#)

Automating IBM SPSS Modeler

Since advanced data mining can be a complex and sometimes lengthy process, IBM® SPSS® Modeler includes several types of coding and automation support.

- **Control Language for Expression Manipulation** (CLEM) is a language for analyzing and manipulating the data that flows along IBM SPSS Modeler streams. Data miners use CLEM extensively in stream operations to perform tasks as simple as deriving profit from cost and revenue data or as complex as transforming web log data into a set of fields and records with usable information.
- **Scripting** is a powerful tool for automating processes in the user interface. Scripts can perform the same kinds of actions that users perform with a mouse or a keyboard. You can also specify output and manipulate generated models.

Understanding data mining

- [Types of models](#)
- [Data Mining Examples](#)

Types of models

IBM® SPSS® Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics. The methods available on the Modeling palette allow you to derive new information from your data and to develop predictive models. Each method has certain strengths and is best suited for particular types of problems.

The *IBM SPSS Modeler Applications Guide* provides examples for many of these methods, along with a general introduction to the modeling process. This guide is available as an online tutorial, and also in PDF format. [More information](#).

Modeling methods are divided into these categories:

- Supervised
- Association
- Segmentation

Supervised Models

Supervised models use the values of one or more **input** fields to predict the value of one or more output, or **target**, fields. Some examples of these techniques are: decision trees (C&R Tree, QUEST, CHAID and C5.0 algorithms), regression (linear, logistic, generalized linear, and Cox regression algorithms), neural networks, support vector machines, and Bayesian networks.

Supervised models help organizations to predict a known result, such as whether a customer will buy or leave or whether a transaction fits a known pattern of fraud. Modeling techniques include machine learning, rule induction, subgroup identification, statistical methods, and multiple model generation.

Supervised nodes

	The Auto Classifier node creates and compares a number of different models for binary outcomes (yes or no, churn or do not churn, and so on), allowing you to choose the best approach for a given analysis. A number of modeling algorithms are supported, making it possible to select the methods you want to use, the specific options for each, and the criteria for comparing the results. The node generates a set of models based on the specified options and ranks the best candidates according to the criteria you specify.
	The Auto Numeric node estimates and compares models for continuous numeric range outcomes using a number of different methods. The node works in the same manner as the Auto Classifier node, allowing you to choose the algorithms to use and to experiment with multiple combinations of options in a single modeling pass. Supported algorithms include neural networks, C&R Tree, CHAID, linear regression, generalized linear regression, and support vector machines (SVM). Models can be compared based on correlation, relative error, or number of variables used.
	The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered “pure” if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).
	The QUEST node provides a binary classification method for building decision trees, designed to reduce the processing time required for large C&R Tree analyses while also reducing the tendency found in classification tree methods to favor inputs that allow more splits. Input fields can be numeric ranges (continuous), but the target field must be categorical. All splits are binary.
	The CHAID node generates decision trees using chi-square statistics to identify optimal splits. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.

	The C5.0 node builds either a decision tree or a rule set. The model works by splitting the sample based on the field that provides the maximum information gain at each level. The target field must be categorical. Multiple splits into more than two subgroups are allowed.
	The Decision List node identifies subgroups, or segments, that show a higher or lower likelihood of a given binary outcome relative to the overall population. For example, you might look for customers who are unlikely to churn or are most likely to respond favorably to a campaign. You can incorporate your business knowledge into the model by adding your own custom segments and previewing alternative models side by side to compare the results. Decision List models consist of a list of rules in which each rule has a condition and an outcome. Rules are applied in order, and the first rule that matches determines the outcome.
	Linear regression models predict a continuous target based on linear relationships between the target and one or more predictors.
	The PCA/Factor node provides powerful data-reduction techniques to reduce the complexity of your data. Principal components analysis (PCA) finds linear combinations of the input fields that do the best job of capturing the variance in the entire set of fields, where the components are orthogonal (perpendicular) to each other. Factor analysis attempts to identify underlying factors that explain the pattern of correlations within a set of observed fields. For both approaches, the goal is to find a small number of derived fields that effectively summarizes the information in the original set of fields.
	The Feature Selection node screens input fields for removal based on a set of criteria (such as the percentage of missing values); it then ranks the importance of remaining inputs relative to a specified target. For example, given a data set with hundreds of potential inputs, which are most likely to be useful in modeling patient outcomes?
	Discriminant analysis makes more stringent assumptions than logistic regression but can be a valuable alternative or supplement to a logistic regression analysis when those assumptions are met.
	Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric range.
	The Generalized Linear model expands the general linear model so that the dependent variable is linearly related to the factors and covariates through a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers the functionality of a wide number of statistical models, including linear regression, logistic regression, loglinear models for count data, and interval-censored survival models.
	A generalized linear mixed model (GLMM) extends the linear model so that the target can have a non-normal distribution, is linearly related to the factors and covariates via a specified link function, and so that the observations can be correlated. Generalized linear mixed models cover a wide variety of models, from simple linear regression to complex multilevel models for non-normal longitudinal data.
	The Cox regression node enables you to build a survival model for time-to-event data in the presence of censored records. The model produces a survival function that predicts the probability that the event of interest has occurred at a given time (t) for given values of the input variables.
	The Support Vector Machine (SVM) node enables you to classify data into one of two groups without overfitting. SVM works well with wide data sets, such as those with a very large number of input fields.
	The Bayesian Network node enables you to build a probability model by combining observed and recorded evidence with real-world knowledge to establish the likelihood of occurrences. The node focuses on Tree Augmented Naïve Bayes (TAN) and Markov Blanket networks that are primarily used for classification.
	The Self-Learning Response Model (SLRM) node enables you to build a model in which a single new case, or small number of new cases, can be used to reestimate the model without having to retrain the model using all data.
	The Time Series node estimates exponential smoothing, univariate Autoregressive Integrated Moving Average (ARIMA), and multivariate ARIMA (or transfer function) models for time series data and produces forecasts of future performance. This Time Series node is similar to the previous Time Series node that was deprecated in SPSS Modeler version 18. However, this newer Time Series node is designed to harness the power of IBM SPSS Analytic Server to process big data, and display the resulting model in the output viewer that was added in SPSS Modeler version 17.
	The k -Nearest Neighbor (KNN) node associates a new case with the category or value of the k objects nearest to it in the predictor space, where k is an integer. Similar cases are near each other and dissimilar cases are distant from each other.
	The Spatio-Temporal Prediction (STP) node uses data that contains location data, input fields for prediction (predictors), a time field, and a target field. Each location has numerous rows in the data that represent the values of each predictor at each time of measurement. After the data is analyzed, it can be used to predict target values at any location within the shape data that is used in the analysis.

Association Models

Association models find patterns in your data where one or more entities (such as events, purchases, or attributes) are associated with one or more other entities. The models construct rule sets that define these relationships. Here the fields within the data can act as both inputs and targets. You could find these associations manually, but association rule algorithms do so much more quickly, and can explore more complex patterns. Apriori and Carma models are examples of the use of such algorithms. One other type of association model is a sequence detection model, which finds sequential patterns in time-structured data.

Association models are most useful when predicting multiple outcomes—for example, customers who bought product X also bought Y and Z. Association models associate a particular conclusion (such as the decision to buy something) with a set of conditions. The advantage of

association rule algorithms over the more standard decision tree algorithms (C5.0 and C&RT) is that associations can exist between any of the attributes. A decision tree algorithm will build rules with only a single conclusion, whereas association algorithms attempt to find many rules, each of which may have a different conclusion.

Association nodes

	The Apriori node extracts a set of rules from the data, pulling out the rules with the highest information content. Apriori offers five different methods of selecting rules and uses a sophisticated indexing scheme to process large data sets efficiently. For large problems, Apriori is generally faster to train; it has no arbitrary limit on the number of rules that can be retained, and it can handle rules with up to 32 preconditions. Apriori requires that input and output fields all be categorical but delivers better performance because it is optimized for this type of data.
	The CARMA model extracts a set of rules from the data without requiring you to specify input or target fields. In contrast to Apriori the CARMA node offers build settings for rule support (support for both antecedent and consequent) rather than just antecedent support. This means that the rules generated can be used for a wider variety of applications—for example, to find a list of products or services (antecedents) whose consequent is the item that you want to promote this holiday season.
	The Sequence node discovers association rules in sequential or time-oriented data. A sequence is a list of item sets that tends to occur in a predictable order. For example, a customer who purchases a razor and aftershave lotion may purchase shaving cream the next time he shops. The Sequence node is based on the CARMA association rules algorithm, which uses an efficient two-pass method for finding sequences.
	The Association Rules Node is similar to the Apriori Node; however, unlike Apriori, the Association Rules Node can process list data. In addition, the Association Rules Node can be used with IBM SPSS Analytic Server to process big data and take advantage of faster parallel processing.

Segmentation Models

Segmentation models divide the data into segments, or clusters, of records that have similar patterns of input fields. As they are only interested in the input fields, segmentation models have no concept of output or target fields. Examples of segmentation models are Kohonen networks, K-Means clustering, two-step clustering and anomaly detection.

Segmentation models (also known as "clustering models") are useful in cases where the specific result is unknown (for example, when identifying new patterns of fraud, or when identifying groups of interest in your customer base). Clustering models focus on identifying groups of similar records and labeling the records according to the group to which they belong. This is done without the benefit of prior knowledge about the groups and their characteristics, and it distinguishes clustering models from the other modeling techniques in that there is no predefined output or target field for the model to predict. There are no right or wrong answers for these models. Their value is determined by their ability to capture interesting groupings in the data and provide useful descriptions of those groupings. Clustering models are often used to create clusters or segments that are then used as inputs in subsequent analyses (for example, by segmenting potential customers into homogeneous subgroups).

Segmentation nodes

	The Auto Cluster node estimates and compares clustering models, which identify groups of records that have similar characteristics. The node works in the same manner as other automated modeling nodes, allowing you to experiment with multiple combinations of options in a single modeling pass. Models can be compared using basic measures with which to attempt to filter and rank the usefulness of the cluster models, and provide a measure based on the importance of particular fields.
	The K-Means node clusters the data set into distinct groups (or clusters). The method defines a fixed number of clusters, iteratively assigns records to clusters, and adjusts the cluster centers until further refinement can no longer improve the model. Instead of trying to predict an outcome, k-means uses a process known as unsupervised learning to uncover patterns in the set of input fields.
	The Kohonen node generates a type of neural network that can be used to cluster the data set into distinct groups. When the network is fully trained, records that are similar should be close together on the output map, while records that are different will be far apart. You can look at the number of observations captured by each unit in the model nugget to identify the strong units. This may give you a sense of the appropriate number of clusters.
	The TwoStep node uses a two-step clustering method. The first step makes a single pass through the data to compress the raw input data into a manageable set of subclusters. The second step uses a hierarchical clustering method to progressively merge the subclusters into larger and larger clusters. TwoStep has the advantage of automatically estimating the optimal number of clusters for the training data. It can handle mixed field types and large data sets efficiently.
	The Anomaly Detection node identifies unusual cases, or outliers, that do not conform to patterns of "normal" data. With this node, it is possible to identify outliers even if they do not fit any previously known patterns and even if you are not exactly sure what you are looking for.

In-Database Mining Models

IBM SPSS Modeler supports integration with data mining and modeling tools that are available from database vendors, including Oracle Data Miner and Microsoft Analysis Services. You can build, score, and store models inside the database—all from within the IBM SPSS Modeler application. For full details, see the *IBM SPSS Modeler In-Database Mining Guide*.

IBM SPSS Statistics Models

If you have a copy of IBM SPSS Statistics installed and licensed on your computer, you can access and run certain IBM SPSS Statistics routines from within IBM SPSS Modeler to build and score models.

Data Mining Examples

The best way to learn about data mining in practice is to start with an example. A number of application examples are available in the *IBM® SPSS® Modeler Applications Guide*, which provides brief, targeted introductions to specific modeling methods and techniques. See the topic [Application examples](#) for more information.

Building streams

- [Stream-building overview](#)
- [Building data streams](#)
- [Tips and Shortcuts](#)

Stream-building overview

Data mining using IBM® SPSS® Modeler focuses on the process of running data through a series of nodes, referred to as a **stream**. This series of nodes represents operations to be performed on the data, while links between the nodes indicate the direction of data flow. Typically, you use a data stream to read data into IBM SPSS Modeler, run it through a series of manipulations, and then send it to a destination, such as a table or a viewer.

For example, suppose that you want to open a data source, add a new field, select records based on values in the new field, and then display the results in a table. In this case, your data stream would consist of four nodes:

	A Variable File node, which you set up to read the data from the data source.
	A Derive node, which you use to add the new, calculated field to the data set.
	A Select node, which you use to set up selection criteria to exclude records from the data stream.
	A Table node, which you use to display the results of your manipulations onscreen.

Building data streams

The unique SPSS® Modeler interface lets you mine your data visually by working with diagrams of data streams. At the most basic level, you can build a data stream using the following steps:

- Add nodes to the stream canvas.
- Connect the nodes to form a stream.
- Specify any node or stream options.
- Run the stream.

This section contains more detailed information on working with nodes to create more complex data streams. It also discusses options and settings for nodes and streams. For step-by-step examples of stream building using the data shipped with SPSS Modeler (in the Demos folder of your program installation), see [Application examples](#).

- [Working with nodes](#)
- [Working with streams](#)
- [Stream descriptions](#)
- [Running streams](#)
- [Working with Models](#)
- [Adding Comments and Annotations to Nodes and Streams](#)
- [Saving data streams](#)
- [Loading files](#)
- [Mapping Data Streams](#)

Working with nodes

Nodes are used in IBM® SPSS® Modeler to help you explore data. Various nodes in the workspace represent different objects and actions. The palette at the bottom of the IBM SPSS Modeler window contains all of the possible nodes used in stream building.

There are several types of nodes. Source nodes bring data into the stream, and are located on the Sources tab of the nodes palette. Process nodes perform operations on individual data records and fields, and can be found in the Record Ops and Field Ops tabs of the palette. Output nodes produce a variety of output for data, charts and model results, and are included on the Graphs, Output and Export tabs of the nodes palette. Modeling nodes use statistical algorithms to create model nuggets, and are located on the Modeling tab, and (if activated) the Database Modeling tab, of the nodes palette. See the topic [Nodes palette](#) for more information.

You connect the nodes to form streams which, when run, let you visualize relationships and draw conclusions. Streams are like scripts--you can save them and reuse them with different data files.

A runnable node that processes stream data is known as a terminal node. A modeling or output node is a terminal node if it is located at the end of a stream or stream branch. You cannot connect further nodes to a terminal node.

Note: You can customize the Nodes palette. See the topic [Customizing the Nodes Palette](#) for more information.

- [Adding Nodes to a Stream](#)
- [Connecting Nodes in a Stream](#)
- [Bypassing Nodes in a Stream](#)
- [Disabling Nodes in a Stream](#)
- [Adding Nodes in Existing Connections](#)
- [Deleting Connections between Nodes](#)
- [Setting options for nodes](#)
- [Caching options for nodes](#)
- [Previewing data in nodes](#)
- [Locking nodes](#)

Adding Nodes to a Stream

There are several ways to add nodes to a stream from the nodes palette:

- Double-click a node on the palette. *Note:* Double-clicking a node automatically connects it to the current stream. See the topic [Connecting Nodes in a Stream](#) for more information.
- Drag and drop a node from the palette to the stream canvas.
- Click a node on the palette, and then click the stream canvas.
- Select an appropriate option from the Insert menu of IBM® SPSS® Modeler.

Once you have added a node to the stream canvas, double-click the node to display its dialog box. The available options depend on the type of node that you are adding. For information about specific controls within the dialog box, click its Help button.

Removing Nodes

To remove a node from the data stream, click it and either press the Delete key, or right-click and select Delete from the menu.

Related information

- [Working with nodes](#)
- [Connecting Nodes in a Stream](#)
- [Bypassing Nodes in a Stream](#)
- [Disabling Nodes in a Stream](#)
- [Adding Nodes in Existing Connections](#)
- [Deleting Connections between Nodes](#)
- [Setting options for nodes](#)
- [Caching options for nodes](#)
- [Previewing data in nodes](#)
- [Locking nodes](#)

Connecting Nodes in a Stream

Nodes added to the stream canvas do not form a data stream until they have been connected. Connections between the nodes indicate the direction of the data as it flows from one operation to the next. There are a number of ways to connect nodes to form a stream: double-clicking, using the middle mouse button, or manually.

To Add and Connect Nodes by Double-Clicking

The simplest way to form a stream is to double-click nodes on the palette. This method automatically connects the new node to the selected node on the stream canvas. For example, if the canvas contains a Database node, you can select this node and then double-click the next node from the palette, such as a Derive node. This action automatically connects the Derive node to the existing Database node. You can repeat this process until you have reached a terminal node, such as a Histogram or Table node, at which point any new nodes will be connected to the last non-terminal node upstream.

To Connect Nodes Using the Middle Mouse Button

On the stream canvas, you can click and drag from one node to another using the middle mouse button. (If your mouse does not have a middle button, you can simulate this by pressing the Alt key while dragging with the mouse from one node to another.)

To Manually Connect Nodes

If you do not have a middle mouse button and prefer to manually connect nodes, you can use the pop-up menu for a node to connect it to another node already on the canvas.

1. Right-click the node from which you want to start the connection. Doing so opens the node menu.
2. On the menu, click Connect.
3. A connection icon is displayed both on the start node and the cursor. Click a second node on the canvas to connect the two nodes.

When connecting nodes, there are several guidelines to follow. You will receive an error message if you attempt to make any of the following types of connections:

- A connection leading to a source node
- A connection leading from a terminal node
- A node having more than its maximum number of input connections
- Connecting two nodes that are already connected
- Circularity (data returns to a node from which it has already flowed)

Related information

- [Working with nodes](#)
- [Adding Nodes to a Stream](#)
- [Bypassing Nodes in a Stream](#)
- [Disabling Nodes in a Stream](#)
- [Adding Nodes in Existing Connections](#)
- [Deleting Connections between Nodes](#)
- [Setting options for nodes](#)
- [Caching options for nodes](#)
- [Previewing data in nodes](#)
- [Locking nodes](#)

Bypassing Nodes in a Stream

When you bypass a node in the data stream, all of its input and output connections are replaced by connections that lead directly from its input nodes to its output nodes. If the node does not have both input and output connections, then all of its connections are deleted rather than rerouted.

For example, you might have a stream that derives a new field, filters fields, and then explores the results in a histogram and table. If you want to also view the same graph and table for data *before* fields are filtered, you can add either new Histogram and Table nodes to the stream, or you can bypass the Filter node. When you bypass the Filter node, the connections to the graph and table pass directly from the Derive node. The Filter node is disconnected from the stream.

To Bypass a Node

1. On the stream canvas, use the middle mouse button to double-click the node that you want to bypass. Alternatively, you can use Alt+double-click.

Note: You can undo this action clicking Undo on the Edit menu or by pressing Ctrl+Z.

Related information

- [Working with nodes](#)
 - [Adding Nodes to a Stream](#)
 - [Connecting Nodes in a Stream](#)
 - [Disabling Nodes in a Stream](#)
 - [Adding Nodes in Existing Connections](#)
 - [Deleting Connections between Nodes](#)
 - [Setting options for nodes](#)
 - [Caching options for nodes](#)
 - [Previewing data in nodes](#)
 - [Locking nodes](#)
-

Disabling Nodes in a Stream

Process nodes with a single input within streams can be disabled, with the result that the node is ignored during running of the stream. This saves you from having to remove or bypass the node and means you can leave it connected to the remaining nodes. You can still open and edit the node settings; however, any changes will not take effect until you enable the node again.

For example, you might have a stream that filters several fields, and then builds models with the reduced data set. If you want to also build the same models *without* fields being filtered, to see if they improve the model results, you can disable the Filter node. When you disable the Filter node, the connections to the modeling nodes pass directly through from the Derive node to the Type node.

To Disable a Node

1. On the stream canvas, right-click the node that you want to disable.
2. Click Disable Node on the pop-up menu.

Alternatively, you can click Node > Disable Node on the Edit menu. When you want to include the node back in the stream, click Enable Node in the same way.

Note: You can undo this action clicking Undo on the Edit menu or by pressing Ctrl+Z.

Related information

- [Working with nodes](#)
 - [Adding Nodes to a Stream](#)
 - [Connecting Nodes in a Stream](#)
 - [Bypassing Nodes in a Stream](#)
 - [Adding Nodes in Existing Connections](#)
 - [Deleting Connections between Nodes](#)
 - [Setting options for nodes](#)
 - [Caching options for nodes](#)
 - [Previewing data in nodes](#)
 - [Locking nodes](#)
-

Adding Nodes in Existing Connections

You can add a new node between two connected nodes by dragging the arrow that connects the two nodes.

1. With the middle mouse button, click and drag the connection arrow into which you want to insert the node. Alternatively, you can hold down the Alt key while clicking and dragging to simulate a middle mouse button.
2. Drag the connection to the node that you want to include and release the mouse button.

Note: You can remove new connections from the node and restore the original by **bypassing** the node.

Related information

- [Working with nodes](#)
- [Adding Nodes to a Stream](#)
- [Connecting Nodes in a Stream](#)
- [Bypassing Nodes in a Stream](#)
- [Disabling Nodes in a Stream](#)
- [Deleting Connections between Nodes](#)
- [Setting options for nodes](#)

- [Caching options for nodes](#)
 - [Previewing data in nodes](#)
 - [Locking nodes](#)
-

Deleting Connections between Nodes

To delete the connection between two nodes:

1. Right-click the connection arrow.
2. On the menu, click Delete Connection.

To delete all connections to and from a node, do one of the following:

- Select the node and press F3.
- Select the node and, on the main menu, click:

Edit > Node > Disconnect

Related information

- [Working with nodes](#)
 - [Adding Nodes to a Stream](#)
 - [Connecting Nodes in a Stream](#)
 - [Bypassing Nodes in a Stream](#)
 - [Disabling Nodes in a Stream](#)
 - [Adding Nodes in Existing Connections](#)
 - [Setting options for nodes](#)
 - [Caching options for nodes](#)
 - [Previewing data in nodes](#)
 - [Locking nodes](#)
-

Setting options for nodes

Once you creat and connect nodes, there are several options for customizing nodes. Right-click a node and select one of the menu options.

- Click Edit to open the dialog box for the selected node.
 - Click Connect to manually connect one node to another.
 - Click Disconnect to delete all links to and from the node.
 - Click Rename and Annotate to open the Annotations tab of the editing dialog box.
 - Click New Comment to add a comment related to the node. See the topic [Adding Comments and Annotations to Nodes and Streams](#) for more information.
 - Click Disable Node to "hide" the node during processing. To make the node visible again for processing, click Enable Node. See the topic [Disabling Nodes in a Stream](#) for more information.
 - Click Cut or Delete to remove the selected node(s) from the stream canvas. *Note:* Clicking Cut allows you to paste nodes, while Delete does not.
 - Click Copy Node to make a copy of the node with no connections. This can be added to a new or existing stream.
 - Click Load Node to open a previously saved node and load its options into the currently selected node. The nodes must be of identical types.
 - Click Retrieve Node to retrieve a node from a connected IBM® SPSS® Collaboration and Deployment Services Repository.
 - Click Save Node to save the node's details in a file. You can load node details only into another node of the same type.
 - Click Store Node to store the selected node in a connected IBM SPSS Collaboration and Deployment Services Repository.
 - Click Cache to expand the menu, with options for caching the selected node.
 - Click Data Mapping to expand the menu, with options for mapping data to a new source or specifying mandatory fields.
 - Click Create SuperNode to expand the menu, with options for creating a SuperNode in the current stream.
 - Click Generate User Input Node to replace the selected node. Examples generated by this node will have the same fields as the current node.
 - Click Run From Here to run all terminal nodes downstream from the selected node.
-

Caching options for nodes

To optimize stream running, you can set up a **cache** on any nonterminal node. When you set up a cache on a node, the cache is filled with the data that passes through the node the next time you run the data stream. From then on, the data is read from the cache (which is stored on disk in a temporary directory) rather than from the data source.

Caching is most useful following a time-consuming operation such as a sort, merge, or aggregation. For example, suppose that you have a source node set to read sales data from a database and an Aggregate node that summarizes sales by location. You can set up a cache on the Aggregate node rather than on the source node because you want the cache to store the aggregated data rather than the entire data set.

Note: Caching at source nodes, which simply stores a copy of the original data as it is read into IBM® SPSS® Modeler, will not improve performance in most circumstances.

Nodes with caching enabled are displayed with a small document icon at the top right corner. When the data is cached at the node, the document icon is green.

To Enable a Cache

1. On the stream canvas, right-click the node and click Cache on the menu.
2. On the caching submenu, click Enable.
3. You can turn the cache off by right-clicking the node and clicking Disable on the caching submenu.

Caching Nodes in a Database

For streams run in a database, data can be cached midstream to a temporary table in the database rather than the file system. When combined with SQL optimization, this may result in significant gains in performance. For example, the output from a stream that merges multiple tables to create a data mining view may be cached and reused as needed. By automatically generating SQL for all downstream nodes, performance can be further improved.

To take advantage of database caching, both SQL optimization and database caching must be enabled. Note that Server optimization settings override those on the Client. See the topic [Setting optimization options for streams](#) for more information.

With database caching enabled, simply right-click any nonterminal node to cache data at that point, and the cache will be created automatically directly in the database the next time the stream is run. If database caching or SQL optimization is not enabled, the cache will be written to the file system instead.

Note: The following databases support temporary tables for the purpose of caching: Db2, Oracle, SQL Server, and Teradata. Other databases, such as Netezza, will use a normal table for database caching. The SQL code can be customized for specific databases - contact Services for assistance.

To Flush a Cache

A white document icon on a node indicates that its cache is empty. When the cache is full, the document icon becomes solid green. If you want to replace the contents of the cache, you must first flush the cache and then re-run the data stream to refill it.

1. On the stream canvas, right-click the node and click Cache on the menu.
2. On the caching submenu, click Flush.

To Save a Cache

You can save the contents of a cache as an IBM SPSS Statistics data file (*.sav). You can then either reload the file as a cache, or you can set up a node that uses the cache file as its data source. You can also load a cache that you saved from another project.

1. On the stream canvas, right-click the node and click Cache on the menu.
2. On the caching submenu, click Save Cache.
3. In the Save Cache dialog box, browse to the location where you want to save the cache file.
4. Enter a name in the File Name text box.
5. Be sure that *.sav is selected in the Files of Type list, and click Save.

To Load a Cache

If you have saved a cache file before removing it from the node, you can reload it.

1. On the stream canvas, right-click the node and click Cache on the menu.
2. On the caching submenu, click Load Cache.
3. In the Load Cache dialog box, browse to the location of the cache file, select it, and click Load.

Previewing data in nodes

To ensure that data is being changed in the way you expect as you build a stream, you could run your data through a Table node at each significant step. To save you from having to do this, you can generate a preview from each node that displays a sample of the data that will be created, thereby reducing the time it takes to build each node.

For nodes upstream of a model nugget, the preview shows the input fields; for a model nugget or nodes downstream of the nugget (except terminal nodes), the preview shows input and generated fields.

The default number of rows displayed is 10; however, you can change this in the stream properties. See the topic [Setting general options for streams](#) for more information.

From the Generate menu, you can create several types of nodes.

Note: When previewing data generated by this node, all property changes will be applied to this node and cannot be cancelled (the same behavior as clicking Apply).

Locking nodes

To prevent other users from amending the settings of one or more nodes in a stream, you can encapsulate the node or nodes in a special type of node called a SuperNode, and then lock the SuperNode by applying password protection.

Working with streams

Once you have connected source, process, and terminal nodes on the stream canvas, you have created a stream. As a collection of nodes, streams can be saved, annotated, and added to projects. You can also set numerous options for streams, such as optimization, date and time settings, parameters, and scripts. These properties are discussed in the topics that follow.

In IBM® SPSS® Modeler, you can use and modify more than one data stream in the same IBM SPSS Modeler session. The right side of the main window contains the managers pane, which helps you to navigate the streams, outputs, and models that are currently open. If you cannot see the managers pane, click Managers on the View menu, then click the Streams tab.

From this tab, you can:

- Access streams.
- Save streams.
- Save streams to the current project.
- Close streams.
- Open new streams.
- Store and retrieve streams from an IBM SPSS Collaboration and Deployment Services repository (if available at your site). See the topic [About the IBM SPSS Collaboration and Deployment Services Repository](#) for more information.

Right-click a stream on the Streams tab to access these options.

- [Setting options for streams](#)
- [Viewing stream operation messages](#)
- [Viewing node execution times](#)
- [Setting Stream and Session Parameters](#)
- [Specifying Runtime Prompts for Parameter Values](#)
- [Specifying Value Constraints for a Parameter Type](#)
- [Stream deployment options](#)
- [Looping Execution for Streams](#)
- [Viewing Global Values for Streams](#)
- [Searching for Nodes in a Stream](#)
- [Renaming streams](#)

Setting options for streams

You can specify a number of options to apply to the current stream. You can also save these options as defaults to apply to all your streams. The options are as follows.

- General. Miscellaneous options such as symbols and text encoding to use in the stream. See [Setting general options for streams](#) for more information.
- Date/Time. Options relating to the format of date and time expressions. See [Setting date and time options for streams](#) for more information.

- Number formats. Options controlling the format of numeric expressions. See [Setting number format options for streams](#) for more information.
- Optimization. Options for optimizing stream performance. See [Setting optimization options for streams](#) for more information.
- Logging and Status. Options controlling SQL logging and record status. See [Setting SQL logging and record status options for streams](#) for more information.
- Layout. Options relating to the layout of the stream on the canvas. See [Setting layout options for streams](#) for more information.
- Analytic Server. Options relating to the use of Analytic Server with SPSS® Modeler. See [Analytic Server stream properties](#) for more information.
- Geospatial. Options relating to the formatting of geospatial data for use in the stream. See [Setting geospatial options for streams](#) for more information.

To set stream options

1. On the File menu, click Stream Properties (or select the stream from the Streams tab in the managers pane, right-click and then click Stream Properties on the pop-up menu).
2. Click the Options tab.

Alternatively, on the Tools menu, click:

Stream Properties > Options

- [Setting general options for streams](#)
- [Setting date and time options for streams](#)
- [Setting number format options for streams](#)
- [Setting optimization options for streams](#)
- [Setting SQL logging and record status options for streams](#)
- [Setting layout options for streams](#)
- [Analytic Server stream properties](#)
- [Setting geospatial options for streams](#)

Setting general options for streams

The general options are a set of miscellaneous options that apply to various aspects of the current stream.

The Basic section includes the following basic options:

- Decimal symbol. Select either a comma (,) or a period (.) as a decimal separator.
- Grouping symbol. For number display formats, select the symbol used to group values (for example, the comma in 3,000.00). Options include none, period, comma, space, and locale-defined (in which case the default for the current locale is used).
- Encoding. Specify the stream default method for text encoding. (Note: Applies to Var. File source node and Flat File export node only. No other nodes use this setting; most data files have embedded encoding information.) You can choose either the system default or UTF-8. The system default is specified in the Windows Control Panel or, if running in distributed mode, on the server computer. See the topic [Unicode Support in IBM SPSS Modeler](#) for more information.
- Ruleset Evaluation. Determines how rule set models are evaluated. By default, rule sets use Voting to combine predictions from individual rules and determine the final prediction. To ensure that rule sets use the first hit rule by default, select First Hit. Note that this option does not apply to Decision List models, which always use the first hit as defined by the algorithm.

Maximum number of rows to show in Data Preview. Specify the number of rows to be shown when a preview of the data is requested for a node. See the topic [Previewing data in nodes](#) for more information.

Maximum members for nominal fields. Select to specify a maximum number of members for nominal (set) fields after which the data type of the field becomes **Typeless**. This option is useful when working with large nominal fields. Note: When the measurement level of a field is set to **Typeless**, its role is automatically set to **None**. This means that the fields are not available for modeling.

Limit set size for Kohonen, and K-Means modeling. Select to specify a maximum number of members for nominal fields used in Kohonen nets and K-Means modeling. The default set size is 20, after which the field is ignored and a warning is raised, providing information on the field in question.

Note that, for compatibility, this option also applies to the old Neural Network node that was replaced in version 14 of IBM® SPSS® Modeler; some legacy streams may still contain this node.

Refresh source nodes on execution. Select to automatically refresh all source nodes when running the current stream. This action is analogous to clicking the Refresh button on a source node, except that this option automatically refreshes all source nodes (except User Input nodes) for the current stream.

Note: Selecting this option flushes the caches of downstream nodes even if the data has not changed. If you use the Run the current stream option from the toolbar, flushing occurs only once per running of the stream, though, which means that you can still use downstream caches as temporary storage for a single running. For example, say that you have set a cache midstream after a complex derive operation and that you have

several graphs and reports attached downstream of this Derive node. When running the stream, the cache at the Derive node will be flushed and refilled but only for the first graph or report. Subsequent terminal nodes will read data from the Derive node cache. Note that if you choose to execute each terminal node individually (when you have more than one terminal node), instead of using the Run the current stream option, cache flushing occurs every time you execute a terminal node.

Display field and value labels in output. Displays field and value labels in tables, charts, and other output. If labels do not exist, the field names and data values will be displayed instead. Labels are turned off by default; however, you can toggle labels on an individual basis elsewhere in IBM SPSS Modeler. You can also choose to display labels on the output window using a toggle button available on the toolbar.

Figure 1. Toolbar icon used to toggle field and value labels



Display execution times. Displays individual execution times for stream nodes on the Execution Times tab after the stream is run. See the topic [Viewing node execution times](#) for more information.

The Automatic Node Creation section includes the following options for creating nodes automatically in individual streams. These options control whether or not to insert the modeling nuggets onto the stream canvas when generating new nuggets. By default, these options only apply to streams created in version 16 or later. In IBM SPSS Modeler 16 or later, if you open a stream created in version 15 or earlier and execute a modeling node, the nugget will not be placed onto the stream canvas as it used to be in previous releases. If you create a new stream using IBM SPSS Modeler 16 or later and execute a modeling node, the nugget generated is placed onto the stream canvas. This is as designed because, for example, the Create model apply nodes for new model output option would likely break pre-16 streams that run in batch, in IBM SPSS Collaboration and Deployment Services, and in other environments where the IBM SPSS Modeler Server client user interface is not present.

- Create model apply nodes for new model output. Automatically creates model apply nodes for the new model output. If you select this option, you can also choose from the Create model update links whether to set the links as enabled, disabled, or not to create them. When a new model applier or source node is created, the link options in the drop-downs control whether the update links between the builder node and the new node are created and, if so, what mode they are in. If links are created, chances are you want them enabled, but these options provide the user with complete control.
- Create source nodes from source builders. Automatically creates source nodes from the source builders. Similar to the previous option, if you select this option you can also choose from the Create source refresh links drop-down whether to set the refresh links as enabled, disabled, or not to create them.

Save As Default. The options specified apply only to the current stream. Click this button to set these options as the default for all streams.

Setting date and time options for streams

These options specify the format to use for various date and time expressions in the current stream.

Import date/time as Select whether to use date/time storage for date/time fields or whether to import them as string variables.

Date format Select a date format to be used for date storage fields or when strings are interpreted as dates by CLEM date functions.

Time format Select a time format to be used for time storage fields or when strings are interpreted as times by CLEM time functions.

Rollover days/mins For time formats, select whether negative time differences are interpreted as referring to the previous day or hour.

Date baseline (1st Jan) Select the baseline years (always 1 January) to be used by CLEM date functions that work with a single date.

2-digit dates start from Specify the cutoff year to add century digits for years that are denoted with only 2 digits. For example, specifying 1930 as the cutoff year assumes that 05/11/02 is in the year 2002. The same setting will use the 20th century for dates after 30; thus 05/11/73 is assumed to be in 1973.

Time zone Select how the time zone is chosen for use with the datetime_now CLEM expression.

- If you select Server, the time zone depends on the following items:
 - If the current stream uses an Analytic Server data source, the datetime_now expression uses the time from the Analytic Server; by default the server uses Coordinated Universal Time time.
 - If the current stream uses a Database source node, the supported databases use SQL pushback, and the datetime_now expression uses the time of the database.
 - For all other streams, the time zone uses the time from SPSS® Modeler Server.
- If you select Modeler client the time zone reflects the time zone details from the machine on which SPSS Modeler is installed.
- Alternatively, you can select any of the Coordinated Universal Time values for the time zone.

Save As Default. The options that are specified apply only to the current stream. Click this button to set these options as the default for all streams.

Related information

- [Working with streams](#)
 - [Setting options for streams](#)
 - [Setting general options for streams](#)
 - [Setting number format options for streams](#)
 - [Setting optimization options for streams](#)
 - [Setting SQL logging and record status options for streams](#)
 - [Setting layout options for streams](#)
 - [Viewing stream operation messages](#)
 - [Viewing node execution times](#)
 - [Setting Stream and Session Parameters](#)
 - [Specifying Runtime Prompts for Parameter Values](#)
 - [Specifying Value Constraints for a Parameter Type](#)
 - [Renaming streams](#)
-

Setting number format options for streams

These options specify the format to use for various numeric expressions in the current stream.

Number display format. You can choose from standard (####.###), scientific (#.###E+##), or currency display formats (\$##.##).

Decimal places (standard, scientific, currency). For number display formats, specifies the number of decimal places to be used when displaying or printing real numbers. This option is specified separately for each display format.

Calculations in. Select Radians or Degrees as the unit of measurement to be used in trigonometric CLEM expressions. See the topic [Trigonometric Functions](#) for more information.

Save As Default. The options specified apply only to the current stream. Click this button to set these options as the default for all streams.

Setting optimization options for streams

You can use the Optimization settings to optimize stream performance. Note that the performance and optimization settings on IBM® SPSS® Modeler Server (if used) override any equivalent settings in the client. If these settings are disabled in the server, then the client cannot enable them. But if they are enabled in the server, the client can choose to disable them.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help>About>Additional Details

If connectivity is enabled, you see the option Server Enablement in the License Status tab.

See [Connecting to IBM SPSS Modeler Server](#) for more information.

Note: Whether SQL pushback and optimization are supported depends on the type of database in use. For the latest information on which databases and ODBC drivers are supported and tested for use with IBM SPSS Modeler, see the corporate Support site at <http://www.ibm.com/support>.

Enable stream rewriting. Select this option to enable stream rewriting in IBM SPSS Modeler. Four types of rewriting are available, and you can select one or more of them. Stream rewriting reorders the nodes in a stream behind the scenes for more efficient operation, without altering stream semantics.

- Optimize SQL generation. This option enables nodes to be reordered within the stream so that more operations can be pushed back using SQL generation for execution in the database. When it finds a node that cannot be rendered into SQL, the optimizer will look ahead to see if there are any downstream nodes that can be rendered into SQL and safely moved in front of the problem node without affecting the stream semantics. Not only can the database perform operations more efficiently than IBM SPSS Modeler, but such pushbacks act to reduce the size of the data set that is returned to IBM SPSS Modeler for processing. This, in turn, can reduce network traffic and speed stream operations. Note that the Generate SQL check box must be selected for SQL optimization to have any effect.
- Optimize CLEM expression. This option enables the optimizer to search for CLEM expressions that can be preprocessed before the stream is run, in order to increase the processing speed. As a simple example, if you have an expression such as `log(salary)`, the optimizer would calculate the actual salary value and pass that on for processing. This can be used both to improve SQL pushback and IBM SPSS Modeler Server performance.
- Optimize syntax execution. This method of stream rewriting increases the efficiency of operations that incorporate more than one node containing IBM SPSS Statistics syntax. Optimization is achieved by combining the syntax commands into a single operation, instead of running each as a separate operation.
- Optimize other execution. This method of stream rewriting increases the efficiency of operations that cannot be delegated to the database. Optimization is achieved by reducing the amount of data in the stream as early as possible. While maintaining data integrity, the

stream is rewritten to push operations closer to the data source, thus reducing data downstream for costly operations, such as joins.

Enable parallel processing. When running on a computer with multiple processors, this option allows the system to balance the load across those processors, which may result in faster performance. Use of multiple nodes or use of the following individual nodes may benefit from parallel processing: C5.0, Merge (by key), Sort, Bin (rank and tile methods), and Aggregate (using one or more key fields).

Generate SQL. Select this option to enable SQL generation, allowing stream operations to be pushed back to the database by using SQL code to generate execution processes, which may improve performance. To further improve performance, Optimize SQL generation can also be selected to maximize the number of operations pushed back to the database. When operations for a node have been pushed back to the database, the node will be highlighted in purple when the stream is run.

- Database caching. For streams that generate SQL to be executed in the database, data can be cached midstream to a temporary table in the database rather than to the file system. When combined with SQL optimization, this may result in significant gains in performance. For example, the output from a stream that merges multiple tables to create a data mining view may be cached and reused as needed. With database caching enabled, simply right-click any nonterminal node to cache data at that point, and the cache is automatically created directly in the database the next time the stream is run. This allows SQL to be generated for downstream nodes, further improving performance. Alternatively, this option can be disabled if needed, such as when policies or permissions preclude data being written to the database. If database caching or SQL optimization is not enabled, the cache will be written to the file system instead. See the topic [Caching options for nodes](#) for more information.
- Use relaxed conversion. This option enables the conversion of data from either strings to numbers, or numbers to strings, if stored in a suitable format. For example, if the data is kept in the database as a string, but actually contains a meaningful number, the data can be converted for use when the pushback occurs.

Note: Due to minor differences in SQL implementation, streams run in a database may return slightly different results from those returned when run in IBM SPSS Modeler. For similar reasons, these differences may also vary depending on the database vendor.

Save As Default. The options specified apply only to the current stream. Click this button to set these options as the default for all streams.

Setting SQL logging and record status options for streams

These settings include various options controlling the display of SQL statements generated by the stream, and the display of the number of records processed by the stream.

Display SQL in the messages log during stream execution. Specifies whether SQL generated while running the stream is passed to the message log.

Display SQL generation details in the messages log during stream preparation. During stream preview, specifies whether a preview of the SQL that would be generated is passed to the messages log.

Display SQL. Specifies whether any SQL that is displayed in the log should contain native SQL functions or standard ODBC functions of the form {fn FUNC (...)} , as generated by SPSS® Modeler. The former relies on ODBC driver functionality that may not be implemented.

Reformat SQL for improved readability. Specifies whether SQL displayed in the log should be formatted for readability.

Show status for records. Specifies when records should be reported as they arrive at terminal nodes. Specify a number that is used for updating the status every N records.

Save As Default. The options specified apply only to the current stream. Click this button to set these options as the default for all streams.

Related information

- [Working with streams](#)
- [Setting options for streams](#)
- [Setting general options for streams](#)
- [Setting date and time options for streams](#)
- [Setting number format options for streams](#)
- [Setting optimization options for streams](#)
- [Setting layout options for streams](#)
- [Viewing stream operation messages](#)
- [Viewing node execution times](#)
- [Setting Stream and Session Parameters](#)
- [Specifying Runtime Prompts for Parameter Values](#)
- [Specifying Value Constraints for a Parameter Type](#)
- [Renaming streams](#)

Setting layout options for streams

These settings provide a number of options relating to the display and use of the stream canvas.

Minimum stream canvas width. Specify the minimum width of the stream canvas in pixels.

Minimum stream canvas height. Specify the minimum height of the stream canvas in pixels.

Stream scroll rate. Specify the scrolling rate for the stream canvas to control how quickly the stream canvas pane scrolls when a node is being dragged from one place to another on the canvas. Higher numbers specify a faster scroll rate.

Icon name maximum. Specify a limit in characters for the names of nodes on the stream canvas.

Icon size. Select an option to scale the entire stream view to one of a number of sizes between 8% and 200% of the standard icon size.

Grid cell size. Select a grid cell size from the list. This number is used for aligning nodes on the stream canvas using an invisible grid. The default grid cell size is 0.25.

Snap to Grid. Select to align icons to an invisible grid pattern (selected by default).

Generated icon placement. Choose where on the canvas to place icons for nodes generated from model nuggets. Default is top left.

Save As Default. The options specified apply only to the current stream. Click this button to set these options as the default for all streams.

Related information

- [Working with streams](#)
- [Setting options for streams](#)
- [Setting general options for streams](#)
- [Setting date and time options for streams](#)
- [Setting number format options for streams](#)
- [Setting optimization options for streams](#)
- [Setting SQL logging and record status options for streams](#)
- [Viewing stream operation messages](#)
- [Viewing node execution times](#)
- [Setting Stream and Session Parameters](#)
- [Specifying Runtime Prompts for Parameter Values](#)
- [Specifying Value Constraints for a Parameter Type](#)
- [Renaming streams](#)

Analytic Server stream properties

These settings provide a number of options for working with Analytic Server.

Maximum number of records to process outside of Analytic Server

Specify the maximum number of records to be imported into SPSS® Modeler server from an Analytic Server data source.

Notification when a node can't be processed in Analytic Server

This setting determines what happens when a stream that would be submitted to Analytic Server contains a node that can't be processed in Analytic Server. Specify whether to issue a warning and continue processing the stream, or throw an error and stop processing.

Split Model Storage Settings

Store split models by reference on Analytic Server when model size (MB) exceeds

Model nuggets are typically stored as part of the stream. Split models with many splits can produce large nuggets, and moving the nugget back and forth between the stream and the Analytic Server can impact performance. As a solution, when a split model exceeds the specified size, it is stored on the Analytic Server, and the nugget in the SPSS Modeler contains a reference to the model.

Default folder to store models by reference on Analytic Server once execution is complete

Specify the default path where you want to store split models on Analytic Server. The path should start with a valid Analytic Server project name.

Folder to store promoted models

Specify the default path where you want to store "promoted" models. A promoted model is not cleaned up when the SPSS Modeler session is over.

Setting geospatial options for streams

Any geospatial field, whether it is a shape, coordinate, or single axis value (such as x or y, or latitude and longitude) has an associated coordinate system. This coordinate system sets attributes such as the origin point (0,0) and the units associated with the values.

There are a number of coordinate systems and there are two types: Geographic and Projected. All of the spatial functions in SPSS® Modeler can only be used with a Projected coordinate system.

Due to the nature of coordinate systems, merging or appending data from two separate geospatial data sources requires that the sources use the same coordinate system. Because of this you must specify a coordinate setting for any geospatial data used in the stream.

Data is automatically reprojected to use the chosen stream coordinate system in the following situations:

- For spatial functions (such as area, closeto, within), the parameter passed to the function is automatically reprojected; however, the original row data is left unchanged.
- When using either the build or scoring (nugget) nodes in Spatio-Temporal Prediction (STP), the location field is automatically reprojected. When scoring, the location that comes out of the nugget is the original location.
- When using the Map Visualization node.

Stream coordinate system. Only available if you select the check box. Click Change to display a list of available Projected Coordinate Systems and select the one you want to use for the current stream.

Save As Default. The coordinate system you select only applies to the current stream. To select the system as the default for all streams, click this button.

- [Selecting geospatial coordinate systems](#)

Related information

- [Working with streams](#)
- [Setting options for streams](#)
- [Setting general options for streams](#)
- [Setting date and time options for streams](#)
- [Setting optimization options for streams](#)
- [Setting SQL logging and record status options for streams](#)
- [Setting layout options for streams](#)
- [Viewing stream operation messages](#)
- [Viewing node execution times](#)
- [Setting Stream and Session Parameters](#)
- [Specifying Runtime Prompts for Parameter Values](#)
- [Specifying Value Constraints for a Parameter Type](#)
- [Renaming streams](#)

Selecting geospatial coordinate systems

All of the spatial functions in SPSS® Modeler can be used only with a Projected coordinate system.

The Select Stream Coordinate System dialog box contains a list of all the projected coordinate systems that you can select for any geospatial data that is used in a stream.

The following information is listed for each coordinate system.

- WKID The Well Known ID that is unique to each coordinate system.
- Name The name of the coordinate system.
- Units The unit of measurement that is associated with the coordinate system.

In addition to the list of all coordinate systems, the dialog box has a Filtering control. If you know all or part of the name of the coordinate system you require, type it in the Name field at the bottom of the dialog box. The list of coordinate systems from which you can choose is automatically filtered to show only the systems with names that contain the text you entered.

Related information

- [Working with streams](#)
- [Setting options for streams](#)
- [Setting general options for streams](#)
- [Setting date and time options for streams](#)
- [Setting optimization options for streams](#)
- [Setting SQL logging and record status options for streams](#)
- [Setting layout options for streams](#)
- [Viewing stream operation messages](#)
- [Viewing node execution times](#)
- [Setting Stream and Session Parameters](#)

- [Specifying Runtime Prompts for Parameter Values](#)
 - [Specifying Value Constraints for a Parameter Type](#)
 - [Renaming streams](#)
-

Viewing stream operation messages

Messages regarding stream operations, such as running, optimization, and time elapsed for model building and evaluation, can easily be viewed using the Messages tab in the stream properties dialog box. Error messages are also reported in this table.

To View Stream Messages

1. On the File menu, click Stream Properties (or select the stream from the Streams tab in the managers pane, right-click and then click Stream Properties on the pop-up menu).
2. Click the Messages tab.

Alternatively, on the Tools menu, click:

Stream Properties > Messages

In addition to messages regarding stream operations, error messages are reported here. When stream running is terminated because of an error, this dialog box will open to the Messages tab with the error message visible. Additionally, the node with errors is highlighted in red on the stream canvas.

If SQL optimization and logging options are enabled in the User Options dialog box, then information on generated SQL is also displayed. See the topic [Setting optimization options for streams](#) for more information.

You can save messages reported here for a stream by clicking Save Messages on the Save button drop-down list (on the left, just below the Messages tab).

You can clear all messages for a given stream by selecting Clear All Messages from the drop-down.

Note that CPU time is the amount of time the server process is utilizing CPU. Elapsed time is the total time between execution start and execution end, so also includes things like transferring files and rendering outputs. CPU time can be more than elapsed time when a stream is leveraging multiple CPUs (parallel execution). When a stream fully pushes back to the database being used as a data source, the CPU time will be zero.

Viewing node execution times

On the Messages tab you can choose to display Execution Times, where you can see the individual execution times for nodes in the stream that run on the IBM® SPSS® Modeler Server. Note that the times may not be accurate for streams run in other areas, such as R or Analytic Server. And the execution time of some nodes cannot be calculated.

Note: For this feature to work, the Display execution times option must be selected on the General setting of the Options tab.

In the table of node execution times, the columns are as follows. Click a column heading to sort the entries into ascending or descending order (for example, to see which nodes have the longest execution times).

Terminal Node. The identifier of the branch to which the node belongs. The identifier is the name of the terminal node at the end of the branch.

Node Label. The name of the node to which the execution time refers.

Node Id. The unique identifier of the node to which the execution time refers. This identifier is generated by the system when the node is created.

Execution Time(s). The time in seconds taken to execute this node. Note that execution time can often vary from the time you see in general messages because time is spent on preparing data and retrieving data in the output when running a stream, and this type of time cannot be calculated.

Setting Stream and Session Parameters

Parameters can be defined for use in CLEM expressions and in scripting. They are, in effect, user-defined variables that are saved and persisted with the current stream, session, or SuperNode and can be accessed from the user interface as well as through scripting. If you save a stream, for example, any parameters set for that stream are also saved. (This distinguishes them from local script variables, which can be used only in the script in which they are declared.) Parameters are often used in scripting to control the behavior of the script, by providing information about fields and values that do not need to be hard coded in the script.

The scope of a parameter depends on where it is set:

- Stream parameters can be set in a stream script or in the stream properties dialog box, and they are available to all nodes in the stream. They are displayed on the Parameters list in the Expression Builder.
- Session parameters can be set in a stand-alone script or in the session parameters dialog box. They are available to all streams used in the current session (all streams listed on the Streams tab in the managers pane).

Parameters can also be set for SuperNodes, in which case they are visible only to nodes encapsulated within that SuperNode.

To Set Stream and Session Parameters through the User Interface

1. To set stream parameters, on the main menu, click:

Tools > Stream Properties > Parameters

2. To set session parameters, click Set Session Parameters on the Tools menu.

Prompt?. Check this box if you want the user to be prompted at runtime to enter a value for this parameter.

Name. Parameter names are listed here. You can create a new parameter by entering a name in this field. For example, to create a parameter for the minimum temperature, you could type minvalue. Do not include the \$P- prefix that denotes a parameter in CLEM expressions. This name is also used for display in the CLEM Expression Builder.

Long name. Lists the descriptive name for each parameter created.

Storage. Select a storage type from the list. Storage indicates how the data values are stored in the parameter. For example, when working with values containing leading zeros that you want to preserve (such as 008), you should select String as the storage type. Otherwise, the zeros will be stripped from the value. Available storage types are string, integer, real, time, date, and timestamp. For date parameters, note that values must be specified using ISO standard notation as shown in the next paragraph.

Value. Lists the current value for each parameter. Adjust the parameter as required. Note that for date parameters, values must be specified in ISO standard notation (that is, **YYYY-MM-DD**). Dates specified in other formats are not accepted.

Type (optional). If you plan to deploy the stream to an external application, select a measurement level from the list. Otherwise, it is advisable to leave the *Type* column as is. If you want to specify value constraints for the parameter, such as upper and lower bounds for a numeric range, select Specify from the list.

Note that long name, storage, and type options can be set for parameters through the user interface only. These options cannot be set using scripts.

Click the arrows at the right to move the selected parameter further up or down the list of available parameters. Use the delete button (marked with an X) to remove the selected parameter.

Related information

- [Working with streams](#)
 - [Setting options for streams](#)
 - [Setting general options for streams](#)
 - [Setting date and time options for streams](#)
 - [Setting number format options for streams](#)
 - [Setting optimization options for streams](#)
 - [Setting SQL logging and record status options for streams](#)
 - [Setting layout options for streams](#)
 - [Viewing stream operation messages](#)
 - [Viewing node execution times](#)
 - [Specifying Runtime Prompts for Parameter Values](#)
 - [Specifying Value Constraints for a Parameter Type](#)
 - [Renaming streams](#)
-

Specifying Runtime Prompts for Parameter Values

If you have streams where you might need to enter different values for the same parameter on different occasions, you can specify runtime prompts for one or more stream or session parameter values.

Parameters. (Optional) Enter a value for the parameter, or leave the default value if there is one.

Turn off these prompts. Select this box if you do not want these prompts to be displayed when you run the stream. You can cause them to be redisplayed by selecting the Prompt? check box on the stream properties or session properties dialog box where the parameters were defined. See the topic [Setting Stream and Session Parameters](#) for more information.

Related information

- [Working with streams](#)
 - [Setting options for streams](#)
 - [Setting general options for streams](#)
 - [Setting date and time options for streams](#)
 - [Setting number format options for streams](#)
 - [Setting optimization options for streams](#)
 - [Setting SQL logging and record status options for streams](#)
 - [Setting layout options for streams](#)
 - [Viewing stream operation messages](#)
 - [Viewing node execution times](#)
 - [Setting Stream and Session Parameters](#)
 - [Specifying Value Constraints for a Parameter Type](#)
 - [Renaming streams](#)
-

Specifying Value Constraints for a Parameter Type

You can make value constraints for a parameter available during stream deployment to an external application that reads data modeling streams. This dialog box allows you to specify the values available to an external user running the stream. Depending on the data type, value constraints vary dynamically in the dialog box. The options shown here are identical to the options available for values from the Type node.

Type. Displays the currently selected measurement level. You can change this value to reflect the way that you intend to use the parameter in IBM® SPSS® Modeler.

Storage. Displays the storage type if known. Storage types are unaffected by the measurement level (continuous, nominal or flag) that you choose for work in IBM SPSS Modeler. You can alter the storage type on the main Parameters tab.

The bottom half of the dialog box dynamically changes depending on the measurement level selected in the Type field.

Continuous Measurement Levels

Lower. Specify a lower limit for the parameter values.

Upper. Specify an upper limit for the parameter values.

Labels. You can specify labels for any value of a range field. Click the Labels button to open a separate dialog box for specifying value labels.

Nominal Measurement Levels

Values. This option allows you to specify values for a parameter that will be used as a nominal field. Values will not be coerced in the IBM SPSS Modeler stream but will be used in a drop-down list for external deployment applications. Using the arrow and delete buttons, you can modify existing values as well as reorder or delete values.

Flag Measurement Levels

True. Specify a flag value for the parameter when the condition is met.

False. Specify a flag value for the parameter when the condition is not met.

Labels. You can specify labels for the values of a flag field.

Related information

- [Working with streams](#)
- [Setting options for streams](#)
- [Setting general options for streams](#)
- [Setting date and time options for streams](#)
- [Setting number format options for streams](#)
- [Setting optimization options for streams](#)
- [Setting SQL logging and record status options for streams](#)
- [Setting layout options for streams](#)
- [Viewing stream operation messages](#)
- [Viewing node execution times](#)
- [Setting Stream and Session Parameters](#)
- [Specifying Runtime Prompts for Parameter Values](#)
- [Renaming streams](#)

Stream deployment options

The Deployment tab of the stream properties dialog box enables you to specify options for deploying the stream within IBM® SPSS® Collaboration and Deployment Services for the purposes of model refresh or automated job scheduling. All streams require a designated scoring branch before they can be deployed. See [Storing and deploying repository objects](#) for more information.

Looping Execution for Streams

Using the Execution tab in the stream properties dialog box, you can set up looping conditions to automate repetitive tasks in the current stream.

Once you set these conditions you can use it as an introduction to scripting as it populates the script window with basic scripting for your stream which you can then modify - perhaps to use as a base from which to build better scripts. See the topic [Global Functions](#) for more information.

To set Looping for a Stream

1. On the File menu, click Stream Properties (or select the stream from the Streams tab in the managers pane, right-click and then click Stream Properties on the pop-up menu).
2. Click the Execution tab.
3. Select the Looping / Conditional Execution execution mode.
4. Click the Looping tab.

Alternatively, on the Tools menu, click:

Stream Properties > Execution

As a further alternative, right click on the node and from the context menu, click:

Looping / Conditional Execution > Edit Looping Settings

Iteration. You cannot edit this row number value, but you can add, delete, or move an iteration up or down using the buttons to the right of the table.

Table headers. These reflect the iteration key and any iteration variables you created when setting up the loop.

Viewing Global Values for Streams

Using the Globals tab in the stream properties dialog box, you can view the global values set for the current stream. Global values are created using a Set Globals node to determine statistics such as mean, sum, or standard deviation for selected fields.

Once the Set Globals node is run, these values are then available for a variety of uses in stream operations. See the topic [Global Functions](#) for more information.

To View Global Values for a Stream

1. On the File menu, click Stream Properties (or select the stream from the Streams tab in the managers pane, right-click and then click Stream Properties on the pop-up menu).
2. Click the Globals tab.

Alternatively, on the Tools menu, click:

Stream Properties > Globals

Globals available. Available globals are listed in this table. You cannot edit global values here, but you can clear all global values for a stream using the Clear All Values button to the right of the table.

Searching for Nodes in a Stream

You can search for nodes in a stream by specifying a number of search criteria, such as node name, category and identifier. This feature can be especially useful for complex streams containing a large number of nodes.

To Search for Nodes in a Stream

1. On the File menu, click Stream Properties (or select the stream from the Streams tab in the managers pane, right-click and then click Stream Properties on the pop-up menu).
2. Click the Search tab.

Alternatively, on the Tools menu, click:

Stream Properties > Search

You can specify more than one option to limit the search, except that searching by node ID (using the ID equals field) excludes the other options.

Node label contains. Check this box and enter all or part of a node label to search for a particular node. Searches are not case-sensitive, and multiple words are treated as a single piece of text.

Node category. Check this box and click a category on the list to search for a particular type of node. Process Node means a node from the Record Ops or Field Ops tab of the nodes palette; Apply Model Node refers to a model nugget.

Keywords include. Check this box and enter one or more complete keywords to search for nodes having that text in the Keywords field on the Annotations tab of the node dialog box. Keyword text that you enter must be an exact match. Separate multiple keywords with semicolons to search for alternatives (for example, entering `proton;neutron` will find all nodes with either of these keywords). See the topic [Annotations](#) for more information.

Annotation contains. Check this box and enter one or more words to search for nodes that contain this text in the main text area on the Annotations tab of the node dialog box. Searches are not case-sensitive, and multiple words are treated as a single piece of text. See the topic [Annotations](#) for more information.

Generates field called. Check this box and enter the name of a generated field (for example, `$C-Drug`). You can use this option to search for modeling nodes that generate a particular field. Enter only one field name, which must be an exact match.

ID equals. Check this box and enter a node ID to search for a particular node with that identifier (selecting this option disables all the preceding options). Node IDs are assigned by the system when the node is created, and can be used to reference the node for the purposes of scripting or automation. Enter only one node ID, which must be an exact match. See the topic [Annotations](#) for more information.

Search in SuperNodes. This box is checked by default, meaning that the search is performed on nodes both inside and outside SuperNodes. Clear the box if you want to perform the search only on nodes outside SuperNodes, at the top level of the stream.

Find. When you have specified all the options you want, click this button to start the search.

Nodes that match the specified options are listed in the lower part of the dialog box. Select a node in the list to highlight it on the stream canvas.

Renaming streams

Using the Annotations tab in the stream properties dialog box, you can add descriptive annotations for a stream and create a custom name for the stream. These options are useful especially when generating reports for streams added to the project pane. See the topic [Annotations](#) for more information.

Stream descriptions

For each stream that you create, IBM® SPSS® Modeler produces a stream description containing information on the contents of the stream. This can be useful if you are trying to see what a stream does but you do not have IBM SPSS Modeler installed, for example when accessing a stream through IBM SPSS Collaboration and Deployment Services.

The stream description is displayed in the form of an HTML document consisting of a number of sections.

General Stream Information

This section contains the stream name, together with details of when the stream was created and last saved.

Description and Comments

This section includes any:

- Stream annotations (see [Annotations](#))
- Comments not connected to specific nodes
- Comments connected to nodes in both the modeling and scoring branches of the stream

Scoring Information

This section contains information under various headings relating to the scoring branch of the stream.

- **Comments.** Includes comments linked only to nodes in the scoring branch.
- **Inputs.** Lists the input fields together with their storage types (for example, string, integer, real and so on).
- **Outputs.** Lists the output fields, including the additional fields generated by the modeling node, together with their storage types.
- **Parameters.** Lists any parameters relating to the scoring branch of the stream and which can be viewed or edited each time the model is scored. These parameters are identified when you click the Scoring Parameters button on the Deployment tab of the stream properties dialog box.
- **Model Node.** Shows the model name and type (for example, Neural Net, C&R Tree and so on). This is the model nugget selected for the Model node field on the Deployment tab of the stream properties dialog box.
- **Model Details.** Shows details of the model nugget identified under the previous heading. Where possible, predictor importance and evaluation charts for the model are included.

Modeling Information

Contains information relating to the modeling branch of the stream.

- **Comments.** Lists any comments or annotations that are connected to nodes in the modeling branch.
- **Inputs.** Lists the input fields together with their role in the modeling branch (in the form of the field role value, for example, Input, Target, Split and so on).
- **Parameters.** Lists any parameters relating to the modeling branch of the stream and which can be viewed or edited each time the model is updated. These parameters are identified when you click the Model Build Parameters button on the Deployment tab of the stream properties dialog box.
- **Modeling node.** Shows the name and type of the modeling node used to generate or update the model.
- [Previewing stream descriptions](#)
- [Exporting Stream Descriptions](#)

Previewing stream descriptions

You can view the contents of a stream description in a web browser by clicking an option on the stream properties dialog box. The contents of the description depend on the options you specify on the Deployment tab of the dialog box. See the topic [Stream Deployment Options](#) for more information.

To view a stream description:

1. On the main IBM® SPSS® Modeler menu, click:
Tools > Stream Properties > Deployment
2. Set the deployment type, the designated scoring node and any scoring parameters.
3. If the deployment type is Model Refresh, you can optionally select a:
 - Modeling node and any model build parameters
 - Model nugget on the scoring branch of the stream
4. Click the Preview Stream Description button.

Exporting Stream Descriptions

You can export the contents of the stream description to an HTML file.

To export a stream description:

1. On the main menu, click:
File > Export Stream Description
2. Enter a name for the HTML file and click Save.

Running streams

Once you specify the required options for streams and connect the required nodes, you can run the stream by running the data through nodes in the stream. There are several ways to run a stream within IBM® SPSS® Modeler. You can:

- Click Run on the Tools menu.
- Click one of the Run... buttons on the toolbar. These buttons allow you to run the entire stream or simply the selected terminal node. See the topic [IBM SPSS Modeler Toolbar](#) for more information.
- Run a single data stream by right-clicking a terminal node and clicking Run on the pop-up menu.
- Run part of a data stream by right-clicking any non-terminal node and clicking Run From Here on the pop-up menu. Doing so causes only those operations after the selected node to be performed.

To halt the running of a stream in progress, you can click the red Stop button on the toolbar, or select Stop Execution from the Tools menu. Note that clicking the Stop button one time tells Modeler to have Modeler Server stop the execution. In some cases execution will stop immediately, but in other cases it must finish the current step before it can stop the entire execution. So time will vary. If you double-click the button, the server connection is dropped and a new connection is established. In most cases, this will close all server processes (and it may take some time). But in some cases, some server process won't be stopped.

If any stream takes longer than three seconds to run, the Execution Feedback dialog box is displayed to indicate the progress.

Some nodes have further displays giving additional information about stream execution. Select the corresponding row in the dialog box to see these displays. The first row is selected automatically.

Working with Models

If a stream includes a modeling node (that is, one from the Modeling or Database Modeling tab of the nodes palette), a **model nugget** is created when the stream is run. A model nugget is a container for a **model**, that is, the set of rules, formulas or equations that enables you to generate predictions against your source data, and which lies at the heart of predictive analytics.

Figure 1. Model nugget



When you successfully run a modeling node, a corresponding model nugget is placed on the stream canvas, where it is represented by a gold diamond-shaped icon (hence the name "nugget"). You can open the nugget and browse its contents to view details about the model. To view the predictions, you attach and run one or more terminal nodes, the output from which presents the predictions in a readable form.

A typical modeling stream consists of two branches. The **modeling branch** contains the modeling node, together with the source and processing nodes that precede it. The **scoring branch** is created when you run the modeling node, and contains the model nugget and the terminal node or nodes that you use to view the predictions.

For more information, see the [IBM® SPSS® Modeler Modeling Nodes](#) guide.

Adding Comments and Annotations to Nodes and Streams

You may need to describe a stream to others in your organization. To help you do this, you can attach explanatory comments to streams, nodes and model nuggets.

Others can then view these comments on-screen, or you can print out an image of the stream that includes the comments.

You can list all the comments for a stream or SuperNode, change the order of comments in the list, edit the comment text, and change the foreground or background color of a comment. See the topic [Listing Stream Comments](#) for more information.

You can also add notes in the form of text annotations to streams, nodes and nuggets by means of the Annotations tab of a stream properties dialog box, a node dialog box, or a model nugget window. These notes are visible only when the Annotations tab is open, except that stream annotations can also be shown as on-screen comments. See the topic [Annotations](#) for more information.

- [Comments](#)
- [Annotations](#)

Related information

- [Comments](#)
- [Operations Involving Comments](#)
- [Listing Stream Comments](#)
- [Converting Annotations to Comments](#)
- [Annotations](#)

Comments

Comments take the form of text boxes in which you can enter any amount of text, and you can add as many comments as you like. A comment can be freestanding (not attached to any stream objects), or it can be connected to one or more nodes or model nuggets in the stream.

Freestanding comments are typically used to describe the overall purpose of the stream; connected comments describe the node or nugget to which they are attached. Nodes and nuggets can have more than one comment attached, and the stream can have any number of freestanding comments.

Note: You can also show stream annotations as on-screen comments, though these cannot be attached to nodes or nuggets. See the topic [Converting Annotations to Comments](#) for more information.

The appearance of the text box changes to indicate the current mode of the comment (or annotation shown as a comment), as the following table shows.

Table 1. Comment and annotation text box modes

Comment text box	Annotation text box	Mode	Indicates	Obtained by...
A yellow text box with a black border and a dashed blue selection outline.	A blue text box with a black border and a dashed blue selection outline.	Edit	Comment is open for editing.	Creating a new comment or annotation, or double-clicking an existing one.
A yellow text box with a black border and a dotted blue selection outline.	A blue text box with a black border and a dotted blue selection outline.	Last selected	Comment can be moved, resized or deleted.	Clicking the stream background after editing, or single-clicking an existing comment or annotation.
A yellow text box with a solid black border and no selection outline.	A blue text box with a solid black border and no selection outline.	View	Editing is complete.	Clicking on another node, comment or annotation after editing.

When you create a new freestanding comment, it is initially displayed in the top left corner of the stream canvas.

If you are attaching a comment to a node or nugget, the comment is initially displayed above the stream object to which it is attached.

The text box is colored white to show that text can be entered. When you have entered the text, click outside the text box. The comment background changes to yellow to show that text entry is complete. The comment remains selected, enabling you to move, resize, or delete it.

When you click again, the border changes to solid lines to show that editing is complete.

Double-clicking a comment changes the text box to edit mode--the background changes to white and the comment text can be edited.

You can also attach comments to SuperNodes.

- [Operations Involving Comments](#)
- [Listing Stream Comments](#)
- [Converting Annotations to Comments](#)

Related information

- [Adding Comments and Annotations to Nodes and Streams](#)
- [Operations Involving Comments](#)
- [Listing Stream Comments](#)
- [Converting Annotations to Comments](#)
- [Annotations](#)

Operations Involving Comments

You can perform a number of operations on comments. You can:

- Add a freestanding comment
- Attach a comment to a node or nugget
- Edit a comment
- Resize a comment
- Move a comment
- Disconnect a comment
- Delete a comment
- Show or hide all comments for a stream

To add a freestanding comment

1. Ensure that nothing is selected on the stream.
2. Do one of the following:

- On the main menu, click:
Insert>New Comment
 - Right-click the stream background and click New Comment on the pop-up menu.
 - Click the New Comment button in the toolbar.
3. Enter the comment text (or paste in text from the clipboard).
 4. Click a node in the stream to save the comment.

To attach a comment to a node or nugget

1. Select one or more nodes or nuggets on the stream canvas.
 2. Do one of the following:
 - On the main menu, click:
Insert>New Comment
 - Right-click the stream background and click New Comment on the pop-up menu.
 - Click the New Comment button in the toolbar.
 3. Enter the comment text.
 4. Click another node in the stream to save the comment.
- Alternatively, you can:
5. Insert a freestanding comment (see previous section).
 6. Do one of the following:
 - Select the comment, press F2, then select the node or nugget.
 - Select the node or nugget, press F2, then select the comment.
 - (Three-button mice only) Move the mouse pointer over the comment, hold down the middle button, drag the mouse pointer over the node or nugget, and release the mouse button.

To attach a comment to an additional node or nugget

If a comment is already attached to a node or nugget, or if it is currently at stream level, and you want to attach it to an additional node or nugget, do one of the following:

- Select the comment, press F2, then select the node or nugget.
- Select the node or nugget, press F2, then select the comment.
- (Three-button mice only) Move the mouse pointer over the comment, hold down the middle button, drag the mouse pointer over the node or nugget, and release the mouse button.

To edit an existing comment

1. Do one of the following:
 - Double-click the comment text box.
 - Select the text box and press Enter.
 - Right-click the text box to display its menu, and click Edit.
2. Edit the comment text. You can use standard Windows shortcut keys when editing, for example Ctrl+C to copy text. Other options during editing are listed in the pop-up menu for the comment.
3. Click outside the text box once to display the resizing controls, then again to complete the comment.

To resize a comment text box

1. Select the comment to display the resizing controls.
2. Click and drag a control to resize the box.
3. Click outside the text box to save the change.

To move an existing comment

If you want to move a comment but not its attached objects (if any), do one of the following:

- Move the mouse pointer over the comment, hold down the left mouse button, and drag the comment to the new position.
- Select the comment, hold down the Alt key, and move the comment using the arrow keys.

If you want to move a comment together with any nodes or nuggets to which the comment is attached:

1. Select all the objects you want to move.
2. Do one of the following:
 - Move the mouse pointer over one of the objects, hold down the left mouse button, and drag the objects to the new position.
 - Select one of the objects, hold down the Alt key, and move the objects using the arrow keys.

To disconnect a comment from a node or nugget

1. Select one or more comments to be disconnected.
2. Do one of the following:

- Press F3.
- Right-click a selected comment and click Disconnect on its menu.

To delete a comment

1. Select one or more comments to be deleted.
2. Do one of the following:
 - Press the Delete key.
 - Right-click a selected comment and click Delete on its menu.

If the comment was attached to a node or nugget, the connection line is deleted as well.

If the comment was originally a stream or SuperNode annotation that had been converted to a freestanding comment, the comment is deleted from the canvas but its text is retained on the Annotations tab for the stream or SuperNode.

To show or hide comments for a stream

1. Do one of the following:
 - On the main menu, click:
View > Comments
 - Click the Show/hide comments button in the toolbar.

Related information

- [Adding Comments and Annotations to Nodes and Streams](#)
 - [Comments](#)
 - [Listing Stream Comments](#)
 - [Converting Annotations to Comments](#)
 - [Annotations](#)
-

Listing Stream Comments

You can view a list of all the comments that have been made for a particular stream or SuperNode.

On this list, you can

- Change the order of comments
- Edit the comment text
- Change the foreground or background color of a comment

Listing Comments

To list the comments made for a stream, do one of the following:

- On the main menu, click:
Tools > Stream Properties > Comments
- Right-click a stream in the managers pane and click Stream Properties, then Comments.
- Right-click a stream background on the canvas and click Stream Properties, then Comments.

Text. The text of the comment. Double-click the text to change the field to an editable text box.

Links. The name of the node to which the comment is attached. If this field is empty, the comment applies to the stream.

Positioning buttons. These move a selected comment up or down in the list.

Comment Colors. To change the foreground or background color of a comment, select the comment, select the Custom colors check box, then select a color from the Background or Foreground list (or both). Click Apply, then click the stream background, to see the effect of the change. Click OK to save the change.

Related information

- [Adding Comments and Annotations to Nodes and Streams](#)
- [Comments](#)
- [Operations Involving Comments](#)
- [Converting Annotations to Comments](#)

- [Annotations](#)
-

Converting Annotations to Comments

Annotations made to streams or SuperNodes can be converted into comments.

In the case of streams, the annotation is converted to a freestanding comment (that is, it is not attached to any nodes) on the stream canvas.

When a SuperNode annotation is converted to a comment, the comment is not attached to the SuperNode on the stream canvas, but is visible when you zoom in to the SuperNode.

To convert a stream annotation to a comment

1. Click Stream Properties on the Tools menu. (Alternatively, you can right-click a stream in the managers pane and click Stream Properties.)
2. Click the Annotations tab.
3. Select the Show annotation as comment check box.
4. Click OK.

To convert a SuperNode annotation to a comment

1. Double-click the SuperNode icon on the canvas.
2. Click the Annotations tab.
3. Select the Show annotation as comment check box.
4. Click OK.

Related information

- [Adding Comments and Annotations to Nodes and Streams](#)
 - [Comments](#)
 - [Operations Involving Comments](#)
 - [Listing Stream Comments](#)
 - [Annotations](#)
-

Annotations

Nodes, streams, and models can be annotated in a number of ways. You can add descriptive annotations and specify a custom name. These options are useful especially when generating reports for streams added to the project pane. For nodes and model nuggets, you can also add ToolTip text to help distinguish between similar nodes on the stream canvas.

Adding annotations

Editing a node or model nugget opens a tabbed dialog box containing an Annotations tab used to set a variety of annotation options. You can also open the Annotations tab directly.

1. To annotate a node or nugget, right-click the node or nugget on the stream canvas and click Rename and Annotate. The editing dialog box opens with the Annotations tab visible.
2. To annotate a stream, click Stream Properties on the Tools menu. (Alternatively, you can right-click a stream in the managers pane and click Stream Properties.) Click the Annotations tab.

Name. Select Custom to adjust the autogenerated name or to create a unique name for the node as displayed on the stream canvas.

Tooltip text. (For nodes and model nuggets only) Enter text used as a tooltip on the stream canvas. This is particularly useful when working with a large number of similar nodes.

Keywords. Specify keywords to be used in project reports and when searching for nodes in a stream, or tracking objects stored in the repository (see [About the IBM SPSS Collaboration and Deployment Services Repository](#)). Multiple keywords can be separated by semicolons--for example, income; crop type; claim value. White spaces at the beginning and end of each keyword are trimmed--for example, income ; crop type will produce the same results as income;crop type. (White spaces within keywords are not trimmed, however. For example, crop type with one space and crop type with two spaces are not the same.)

The main text area can be used to enter lengthy annotations regarding the operations of the node or decisions made in the node. For example, when you are sharing and reusing streams, it is helpful to take notes on decisions such as discarding a field with numerous blanks using a Filter node. Annotating the node stores this information with the node. You can also choose to include these annotations in a project report created from the project pane. See the topic [Introduction to Projects](#) for more information.

Show annotation as comment. (For stream and SuperNode annotations only) Check this box to convert the annotation to a freestanding comment that will be visible on the stream canvas. See the topic [Adding Comments and Annotations to Nodes and Streams](#) for more information.

ID. Displays a unique ID that can be used to reference the node for the purpose of scripting or automation. This value is automatically generated when the node is created and will not change. Also note that to avoid confusion with the letter "O," zeros are not used in node IDs. Use the copy button at the right to copy and paste the ID into scripts or elsewhere as needed.

Saving data streams

After you create a stream, you can save it for future reuse.

To save a stream

1. On the File menu, click Save Stream or Save Stream As.
2. In the Save dialog box, browse to the folder in which you want to save the stream file.
3. Enter a name for the stream in the File Name text box.
4. Select Add to project if you would like to add the saved stream to the current project.

Clicking Save stores the stream with the extension *.str in the specified directory.

Automatic backup files. Each time a stream is saved, the previously saved version of the file is automatically preserved as a backup, with a hyphen appended to the filename (for example mystream.str-). To restore the backed-up version, simply delete the hyphen and reopen the file.

- [Saving States](#)
- [Saving Nodes](#)
- [Saving multiple stream objects](#)
- [Saving Output](#)
- [Encrypting and Decrypting Information](#)

Saving States

In addition to streams, you can save **states**, which include the currently displayed stream diagram and any model nuggets that you have created (listed on the Models tab in the managers pane).

To Save a State

1. On the File menu, click:
State... Save State or Save State As
2. In the Save dialog box, browse to the folder in which you want to save the state file.

Clicking Save stores the state with the extension *.cst in the specified directory.

Saving Nodes

You can also save an individual node by right-clicking the node on the stream canvas and clicking Save Node on the pop-up menu. Use the file extension *.nod.

Saving multiple stream objects

When you exit IBM® SPSS® Modeler with multiple unsaved objects, such as streams, projects, or model nuggets, you will be prompted to save before completely closing the software. If you choose to save items, a dialog box displays options for saving each object.

1. Simply select the check boxes for the objects that you want to save.
2. Click OK to save each object in the required location.

You will then be prompted with a standard Save dialog box for each object. After you finish saving, the application will close.

Saving Output

Tables, graphs, and reports generated from IBM® SPSS® Modeler output nodes can be saved in output object (*.cou) format.

1. When viewing the output you want to save, on the output window menus click:
`File > Save`
2. Specify a name and location for the output file.
3. Optionally, select Add file to project in the Save dialog box to include the file in the current project. See the topic [Introduction to Projects](#) for more information.

Alternatively, you can right-click any output object listed in the managers pane and select Save from the pop-up menu.

Encrypting and Decrypting Information

When you save a stream, node, project, output file, or model nugget, you can encrypt it to prevent its unauthorized use. To do this, you select an extra option when saving, and add a password to the item being saved. This encryption can be set for any of the items that you save and adds extra security to them; it is not the same as the SSL encryption used if you are passing files between IBM® SPSS® Modeler and IBM SPSS Modeler Server.

When you try to open an encrypted item, you are prompted to enter the password. After you enter the correct password, the item is decrypted automatically and opens as usual.

To Encrypt an Item

1. In the Save dialog box, for the item to be encrypted, click Options. The Encryption Options dialog box opens.
2. Select Encrypt this file.
3. Optionally, for further security, select Mask password. This displays anything you enter as a series of dots.
4. Enter the password. *Warning:* If you forget the password, the file or model cannot be opened.
5. If you selected Mask password, re-enter the password to confirm that you entered it correctly.
6. Click OK to return to the Save dialog box.

Note: If you save a copy of any encryption-protected item, the new item is automatically saved in an encrypted format using the original password unless you change the settings in the Encryption Options dialog box.

Loading files

You can reload a number of saved objects in IBM® SPSS® Modeler:

- Streams (.str)
- States (.cst)
- Models (.gm)
- Models palette (.gen)
- Nodes (.nod)
- Output (.cou)
- Projects (.cpj)

Opening new files

Streams can be loaded directly from the File menu.

- On the File menu, click Open Stream.

All other file types can be opened using the submenu items available on the File menu. For example, to load a model, on the File menu click:

`Models > Open Model` or `Load Models Palette`

Opening recently used files

For quick loading of recently used files, you can use the options at the bottom of the File menu.

Select Recent Streams, Recent Projects, or Recent States to expand a list of recently used files.

Mapping Data Streams

Using the mapping tool, you can connect a new data source to a preexisting stream. The mapping tool will not only set up the connection but it will also help you to specify how fields in the new source will replace those in the existing stream. Instead of re-creating an entire data stream for a new data source, you can simply connect to an existing stream.

The data mapping tool allows you to join together two stream fragments and be sure that all of the (essential) field names match up properly. In essence, mapping data results simply in the creation of a new Filter node, which matches up the appropriate fields by renaming them.

There are two equivalent ways to map data:

Select replacement node. This method starts with the node to be replaced. First, you right-click the node to replace; then, using the Data Mapping > Select Replacement Node option from the pop-up menu, select the node with which to replace it.

Map to. This method starts with the node to be introduced to the stream. First, right-click the node to introduce; then, using the Data Mapping > Map To option from the pop-up menu, select the node to which it should join. This method is particularly useful for mapping to a terminal node. Note: You cannot map to Merge or Append nodes. Instead, you should simply connect the stream to the Merge node in the normal manner.

Data mapping is tightly integrated into stream building. If you try to connect to a node that already has a connection, you will be offered the option of replacing the connection or mapping to that node.

- [Mapping Data to a Template](#)
- [Mapping between Streams](#)
- [Specifying Essential Fields](#)
- [Examining Mapped Fields](#)

Related information

- [Mapping Data to a Template](#)
- [Mapping between Streams](#)
- [Specifying Essential Fields](#)
- [Examining Mapped Fields](#)

Mapping Data to a Template

To replace the data source for a template stream with a new source node bringing your own data into IBM® SPSS® Modeler, you should use the Select Replacement Node option from the Data Mapping pop-up menu. This option is available for all nodes except Merge, Aggregate, and all terminal nodes. Using the data mapping tool to perform this action helps ensure that fields are matched properly between the existing stream operations and the new data source. The following steps provide an overview of the data mapping process.

Step 1: Specify essential fields in the original source node. In order for stream operations to run properly, essential fields should be specified. See the topic [Specifying Essential Fields](#) for more information.

Step 2: Add new data source to the stream canvas. Using one of the source nodes, bring in the new replacement data.

Step 3: Replace the template source node. Using the Data Mapping option on the pop-up menu for the template source node, click Select Replacement Node, then select the source node for the replacement data.

Step 4: Check mapped fields. In the dialog box that opens, check that the software is mapping fields properly from the replacement data source to the stream. Any unmapped essential fields are displayed in red. These fields are used in stream operations and must be replaced with a similar field in the new data source in order for downstream operations to function properly. See the topic [Examining Mapped Fields](#) for more information.

After using the dialog box to ensure that all essential fields are properly mapped, the old data source is disconnected and the new data source is connected to the stream using a Filter node called *Map*. This Filter node directs the actual mapping of fields in the stream. An *Unmap* Filter node is also included on the stream canvas. The *Unmap* Filter node can be used to reverse field name mapping by adding it to the stream. It will undo the mapped fields, but note that you will have to edit any downstream terminal nodes to reselect the fields and overlays.

Related information

- [Mapping Data Streams](#)
- [Mapping between Streams](#)
- [Specifying Essential Fields](#)
- [Examining Mapped Fields](#)

Mapping between Streams

Similar to connecting nodes, this method of data mapping does not require you to set essential fields beforehand. With this method, you simply connect from one stream to another using Map to from the Data Mapping pop-up menu. This type of data mapping is useful for mapping to terminal nodes and copying and pasting between streams. *Note:* Using the Map to option, you cannot map to Merge, Append, and all types of source nodes.

To Map Data between Streams

1. Right-click the node that you want to use for connecting to the new stream.
2. On the menu, click:
 Data Mapping > Map to
3. Use the cursor to select a destination node on the target stream.
4. In the dialog box that opens, ensure that fields are properly matched and click OK.

Related information

- [Mapping Data Streams](#)
- [Mapping Data to a Template](#)
- [Specifying Essential Fields](#)
- [Examining Mapped Fields](#)

Specifying Essential Fields

When mapping to an existing stream, essential fields will typically be specified by the stream author. These essential fields indicate whether a particular field is used in downstream operations. For example, the existing stream may build a model that uses a field called *Churn*. In this stream, *Churn* is an essential field because you could not build the model without it. Likewise, fields used in manipulation nodes, such as a Derive node, are necessary to derive the new field. Explicitly setting such fields as essential helps to ensure that the proper fields in the new source node are mapped to them. If mandatory fields are not mapped, you will receive an error message. If you decide that certain manipulations or output nodes are unnecessary, you can delete the nodes from the stream and remove the appropriate fields from the Essential Fields list.

To Set Essential Fields

1. Right-click the source node of the template stream that will be replaced.
2. On the menu, click:
 Data Mapping > Specify Essential Fields
3. Using the Field Chooser, you can add or remove fields from the list. To open the Field Chooser, click the icon to the right of the fields list.

Related information

- [Mapping Data Streams](#)
- [Mapping Data to a Template](#)
- [Mapping between Streams](#)
- [Examining Mapped Fields](#)

Examining Mapped Fields

Once you have selected the point at which one data stream or data source will be mapped to another, a dialog box is displayed for you to select fields for mapping or to ensure that the system default mapping is correct. If essential fields have been set for the stream or data source and they are unmatched, these fields are displayed in red. Any unmapped fields from the data source will pass through the Filter node unaltered, but note that you can map non-essential fields as well.

Original. Lists all fields in the template or existing stream—all of the fields that are present further downstream. Fields from the new data source will be mapped to these fields.

Mapped. Lists the fields selected for mapping to template fields. These are the fields whose names may have to change to match the original fields used in stream operations. Click in the table cell for a field to activate a list of available fields.

If you are unsure of which fields to map, it may be useful to examine the source data closely before mapping. For example, you can use the Types tab in the source node to review a summary of the source data.

Related information

- [Mapping Data Streams](#)
 - [Mapping Data to a Template](#)
 - [Mapping between Streams](#)
 - [Specifying Essential Fields](#)
-

Tips and Shortcuts

Work quickly and easily by familiarizing yourself with the following shortcuts and tips:

- **Build streams quickly by double-clicking.** Simply double-click a node on the palette to add and connect it to the current stream.
- **Use key combinations to select downstream nodes.** Press Ctrl+Q and Ctrl+W to toggle the selection of all nodes downstream.
- **Use shortcut keys to connect and disconnect nodes.** When a node is selected in the canvas, press F2 to begin a connection, press Tab to move to the required node, and press Shift+Spacebar to complete the connection. Press F3 to disconnect all inputs and outputs to the selected node.
- **Customize the Nodes Palette tab with your favorite nodes.** On the Tools menu, click Manage Palettes to open a dialog box for adding, removing, or moving the nodes shown on the Nodes Palette.
- **Rename nodes and add ToolTips.** Each node dialog box includes an Annotations tab on which you can specify a custom name for nodes on the canvas as well as add ToolTips to help organize your stream. You can also include lengthy annotations to track progress, save process details, and denote any business decisions required or achieved.
- **Insert values automatically into a CLEM expression.** Using the Expression Builder, accessible from a variety of dialog boxes (such as those for Derive and Filler nodes), you can automatically insert field values into a CLEM expression. Click the values button on the Expression Builder to choose from existing field values.

Figure 1. Values button



- **Browse for files quickly.** When browsing for files on an Open dialog box, use the File list (click the yellow diamond button at the top of the dialog box, next to the Look In field) to access previously used directories as well as IBM® SPSS® Modeler default directories. Use the forward and back buttons to scroll through accessed directories.
- **Minimize output window clutter.** You can close and delete output quickly using the red X button at the top right corner of all output windows. This enables you to keep only promising or interesting results on the Outputs tab of the managers pane.

A full range of keyboard shortcuts is available for the software. See the topic [Keyboard Accessibility](#) for more information.

Did you know that you can...

- Drag and select a group of nodes on the stream canvas using your mouse.
- Copy and paste nodes from one stream to another.
- Access Help from every dialog box and output window.
- Get Help on CRISP-DM, the Cross-Industry Standard Process for Data Mining. (On the Help menu, click CRISP-DM Help.)

Related information

- [Using shortcut keys](#)
-

Building charts

Required service

Watson Studio

Data format

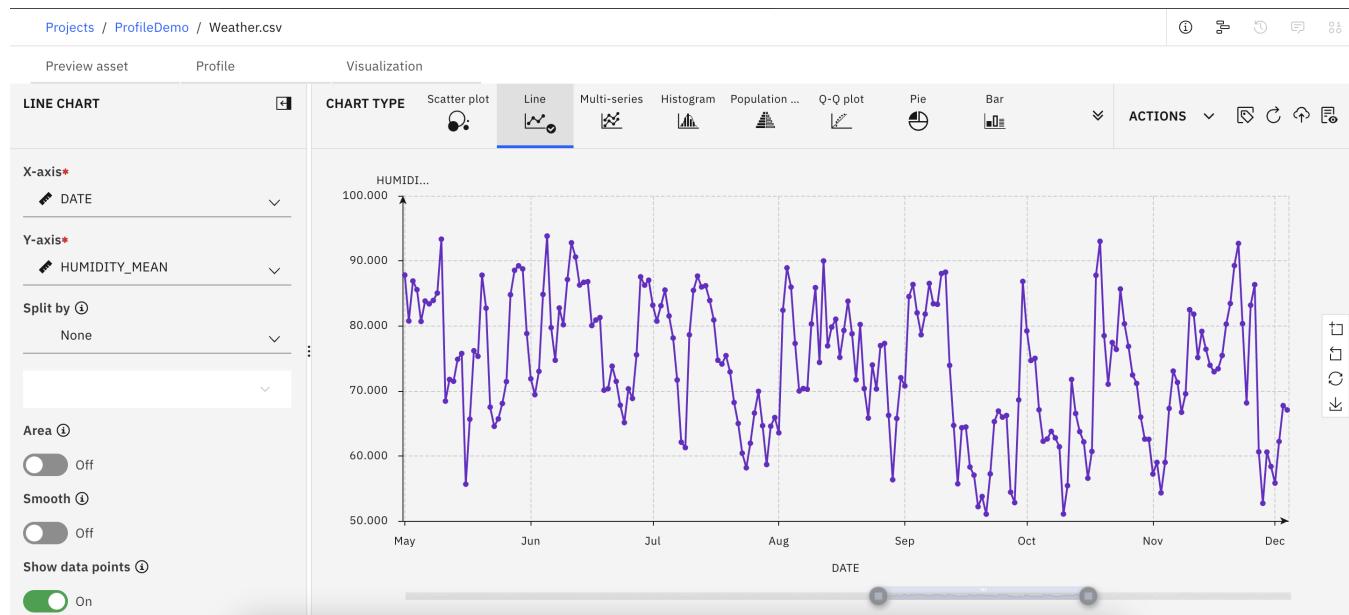
Tabular: Avro, CSV, JSON, Parquet, TSV, SAV, Microsoft Excel .xls and .xlsx files, SAS, delimited text files, and connected data.

For more information on supported data sources, see [Connection types](#).

Data size

No limit

You can create graphics similar to the following example that shows how humidity values over time.



Starting the Chart builder

- Right-click the data node that you want to work with and then select View Data.
- [Chart catalog](#)
- [Layout and terms](#)
- [Building a chart from the chart type gallery](#)
- [Chart types](#)
- [Global visualization preferences](#)

Chart catalog

The Chart catalog provides options for finding suitable charts by browsing chart categories, using keyword search, or filtering by chart characteristics. The Chart catalog can be accessed from any active dataset tab.

From any active dataset tab click the Chart builder control that is available in the toolbar. Alternately, access the catalog through Visualize > Chart builder.... There are various methods you can use to search for suitable chart types.

- The Find chart type field can be used to search for specific charts and keywords. As you enter a chart name or keywords, the number of available chart tiles filters to provide results for only the specified chart names or keywords.
Note: You can use the tile view and list view icons next to the Find chart type field to control how chart types are presented in the Chart catalog.
- The Chart catalog tiles are grouped according to the their corresponding categories. You can scroll through the available tiles until you locate the desired chart type.
- The Filter pane provides options for filtering charts according to their corresponding categories, purpose, dependent variables, and statistics.
Notes:
 - The number next to each Filter pane heading identifies the number of filters that are currently selected for each section.
 - Click Reset filter to clear the Filter pane settings.

Layout and terms

Canvas

The canvas is the area of the Chart Builder dialog where you build the chart.

Chart type

Lists the available chart types. The graphic elements are the items in the chart that represent data (bars, points, lines, and so on).

Details pane

The Details pane provides the basic chart building blocks.

Chart settings

Provides options for selecting which variables are used to build the chart, distribution method, title and subtitle fields, and so on. Depending on the selected chart type, the Details pane options might vary. For more information, see [Chart types](#).

Actions

Provides options for downloading chart configuration files, downloading charts as image files, resetting charts, and setting the global chart preferences.

Building a chart from the chart type gallery

Use chart type gallery for building charts. Following are general steps for building a chart from the gallery. For more information, see [Chart types](#).

1. In the Chart Type section, select a chart category. A preview version of the selected chart type is shown on the chart canvas.
2. If the canvas already displays a chart, the new chart replaces the chart's axis set and graphic elements.
 - a. Depending on the selected chart type, the available variables are presented under a number of different headings in the Details pane (for example, Category for bar charts, X-axis and Y-axis for line charts). Select the appropriate variables for the selected chart type.

Chart types

The gallery contains a collection of the most commonly used charts.

- [3D charts](#)
- [Bar charts](#)
- [Box plots](#)
- [Bubble charts](#)
- [Candlestick charts](#)
- [Circle packing charts](#)
- [Custom charts](#)
- [Dendrogram charts](#)
- [Dual Y-axes charts](#)
- [Error bar charts](#)
- [Evaluation charts](#)
- [Heat map charts](#)
- [Histogram charts](#)
- [Line charts](#)
- [Map charts](#)
- [Math curve charts](#)
- [Multi-chart charts](#)
- [Multiple series charts](#)
- [Parallel charts](#)
- [Pareto charts](#)
- [Pie charts](#)
- [Population pyramid charts](#)
- [Q-Q plots](#)
- [Radar charts](#)
- [Relationship charts](#)
- [Scatter plots and dot plots](#)
- [Scatter matrix charts](#)
- [Series array charts](#)
- [Sunburst charts](#)
- [t-SNE charts](#)
- [Time plots](#)
- [Theme River charts](#)
- [Tree charts](#)
- [Treemap charts](#)
- [Word cloud charts](#)

3D charts

3D charts are commonly used to represent multiple-variable functions and include a z-axis variable that is a function of both the x and y-axis variables.

Creating a simple 3D chart

1. In the Chart Type section, click the 3D icon.

3D



The canvas updates to display a 3D chart template.

2. Select the chart Type from the drop-down list.
3. Select an X-axis variable from the drop-down list.
4. Select an Y-axis variable from the drop-down list.
5. Select an Z-axis variable from the drop-down list.

Options

Type

Lists the chart types that are available to represent the data.

X-axis

Lists variables that are available for the chart's X-axis.

Y-axis

Lists variables that are available for the chart's Y-axis.

Z-axis

Lists variables that are available for the chart's Z-axis.

Tooltip info

Lists the variables that can be used to generate tooltip information when the cursor hovers over a data point.

Color map

Lists available color map variables. These variables use color progression, based on the range of values in the specified column, to represent themselves in the plot points. Color maps are also known as choropleth maps.

Size map

Lists available size map variables. These variables use differing sizes to represent themselves in the plot points.

Z ratio

Sets the scale of the Z-axis data values, relative to the X and Y axes.

Rotate

Enables and disables chart rotation.

Data point tooltips

Controls where the data point tooltips display (right of data points, top-right of chart, or hide).

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Bar charts

Bar charts are useful for summarizing categorical variables. For example, you can use a bar chart to show the number of men and the number of women who participated in a survey. You can also use a bar chart to show the mean salary for men and the mean salary for women.

Creating a simple bar chart

1. In the Chart Type section, click the Bar icon.

Bar



The canvas updates to display a bar chart template.

2. Select a categorical (nominal or ordinal) variable as the Category variable. You can use a scale variable, but the results are useful in only a few special cases. A bar chart looks best with a limited number of distinct values. If you create a bar chart with a scale Category axis, the bars are thin because each bar is drawn at an exact value, and the bar cannot overlap other continuous values.
3. Select a statistic from the Summary list. The result of any statistic determines the height of the bars. If the statistic you want does not appear in the Summary list, it might require a variable. Select a variable from the Value list and check if the statistic is now available. Other chart type limitations might exist. For example, error bar charts can be calculated only for specific statistics.

Options

Category

Lists variables that are available for the chart's X-axis.

Order based on

Select a sorting option for the categories within the variable.

Category name

Use the category labels for sorting the variable's categories. The labels appear in the chart, usually as tick or legend labels.

Category value

Use the value that is stored in the data set for sorting the variable's categories. The category's value is what identifies the category in the data set. It often differs from its label and is not necessarily descriptive. For example, the value might be a number (for example, 1), while the label is a text description of the category (for example, *Female*).

Category order

Select the order in which variable categories are sorted.

As read

Variable categories are presented as they appear in the data set.

Ascending

Sort variable categories in ascending order.

Descending

Sort variable categories in descending order.

Summary

Select a statistical summary function for the graphic element. The result of the statistic determines the position of the graphic elements on the Y-axis. In a 2-D chart, the statistic is calculated for each value on the X-axis. In a 3-D chart, it is calculated for the intersection of values on the X-axis and Z-axis.

Two types of statistical summary functions are available. The distinction is important because it determines whether you need to specify a Value variable.

- **Functions that do not require a value variable.** Functions that do not require a variable. All count and percentage statistics are in this category. These statistics are available when the Value variable is not defined.
- **Functions that do require a value variable.** Functions that do require a Value variable. For example, the **Mean** function requires a variable on which the mean is calculated. These statistics are available when the Value variable is not defined.

Value

This field displays when a Summary function that requires a value variable, is selected. Select a variable to serve as the value.

Split by

Select a categorical variable that creates a table of charts, with a cell for each category in the Split by variable. Like grouping, split by variables essentially add more dimensions to your chart by displaying information for each variable category. For more information, see [Adding Split by variables](#).

Split type

When a Split by variable is selected, you can choose to display the resulting category bars as either stacked or clustered. Clustering and stacking add dimensionality within the chart. Clustering splits one bar into multiple bars, and stacking creates segments in each bar. Be careful that you choose the right statistic for stacking. When the values are added (stacked), the result must make sense. For example, adding and stacking mean (averaged) values is not usually meaningful.

Bar type

Select the bar chart type from the provided options.

- X-axis
- Y-axis
- X-axis inverse
- Y-axis inverse
- Polar-angle axis
- Polar-radius axis
- Polar-rainbow

Label position

Select the chart's label position from the drop-down menu.

- none
- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft

- insideTopRight
- insideBottomRight

Show reference line

The toggle control enables and disables the display of reference lines in the chart. Available options are Min, Max, and Average, which display reference lines at the chart's minimum, maximum and average values.

Enter a reference line value

When Show reference line is enabled, this setting provides the option of specifying a reference line value. Click Add another column to specify more reference line values.

Transpose

When enabled, the chart's X and Y axes are transposed.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

XAxis label

The x-axis label, which is placed beneath the x-axis.

YAxis label

The y-axis label, which is placed above the y-axis.

Box plots

A box plot chart shows the five statistics (minimum, first quartile, median, third quartile, and maximum). It is useful for displaying the distribution of a scale variable and pinpointing outliers.

Creating a simple box plot

1. In the Chart Type section, click the Box plot icon.

Box plot



The canvas updates to display a box plot chart template.

2. Select one or more scale variables as the Columns variable.

Note: The statistic for a dot plot is Box plot. You cannot change this setting.

Options

Columns

Lists variables that are available for the chart's X-axis.

Click Add another column to add more columns.

Split by

Select a categorical variable that creates a table of charts, with a cell for each category in the Split by variable. Like grouping, split by variables essentially add more dimensions to your chart by displaying information for each variable category. For more information, see [Adding Split by variables](#).

Category order

Select the order in which variable categories are sorted.

As read

Variable categories are presented as they appear in the data set.

Ascending

Sort variable categories in ascending order.

Descending

Sort variable categories in descending order.

Strength holder of IQR

The strength holder of the inter-quartile range (**N*IQR**). The **N** default value is 1.5.

Normalize data

When enabled, this setting transforms data into a normal distribution compares data from multiple data sets or multiple columns. This setting creates 100% stacking for counts and converts statistics to percents.

Transpose

When enabled, the chart's X and Y axes are transposed.

Show the outlier

When enabled, outliers display on the chart.

Show extreme outlier style
When enabled, extreme outlier style displays on the chart.

Show label
When enabled, column labels display on the chart. Only scatter series data is supported.

Primary title
The chart title.

Subtitle
The chart subtitle, which is placed directly beneath the chart title.

Footnote
The chart footnote, which is placed beneath the chart.

XAxis label
The x-axis label, which is placed beneath the x-axis.

YAxis label
The y-axis label, which is placed above the y-axis.

Bubble charts

Bubble charts display categories in your groups as nonhierarchical packed circles. The size of each circle (bubble) is proportional to its value. Bubble charts are useful for comparing relationships in your data.

Creating a simple bubble chart

1. In the Chart Type section, click the Bubble icon.

Bubble



The canvas updates to display a bubble chart template.

2. Select a Columns variable from the drop-down list.
Note: Click Add another column to include more column variables.

Options

Columns
Lists variables that are available for the chart.
Click Add another column to add more columns.

Group color
Turn on or off color groupings.

Primary title
The chart title.

Subtitle
The chart subtitle, which is placed directly beneath the chart title.

Footnote
The chart footnote, which is placed beneath the chart.

Candlestick charts

Candlestick charts are a style of financial charts that are used to describe price movements of a security, derivative, or currency. Each candlestick element typically shows one day. A one-month chart might show the 20 trading days as 20 candlesticks elements. Candlestick charts are most often used in the analysis of equity and currency price patterns and are similar to box plots.

The data set that is used to create a candlestick chart must contain open, high, low, and close values for each time period you want to display.

Creating a simple Candlestick chart

1. In the Chart Type section, click the Candlestick icon.

Candlestick



The canvas updates to display a Candlestick chart template.

2. Select a variable as the X-axis variable.
3. Select a variable as the High variable.
4. Select a variable as the Low variable.

Options

X-axis

Lists variables that are available for the chart's X-axis.

Summary

When enabled, summary calculations for available for the following options.

High

Lists variables that are available for the chart's high price value.

High field summary

Select a statistical summary function for the selected high variable.

Low

Lists variables that are available for the chart's low price value.

Open

Lists variables that are available for the chart's opening price value.

Close

Lists variables that are available for the chart's closing price value.

Volume

Lists variables that are available for the chart's volume bars.

Category order

Select the order in which variable categories are sorted.

As read

Variable categories are presented as they appear in the data set.

Ascending

Sort variable categories in ascending order.

Descending

Sort variable categories in descending order.

Candlestick

Toggles the chart data to display as either candlestick or line.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

XAxis label

The x-axis label, which is placed beneath the x-axis.

YAxis label

The y-axis label, which is placed above the y-axis.

Circle packing charts

Circle packing charts display hierarchical data as a set of nested areas to visualize a large amount of hierarchically structured data. It's similar to a treemap, but uses circles instead of rectangles. Circle packing charts use containment (nesting) to display hierarchy data.

Creating a simple circle packing chart

1. In the Chart Type section, click the Time plot icon.

Circle packing



The canvas updates to display a circle packing chart template.

2. Select a Columns variable from the drop-down list.

Note: Click Add another column to include more column variables.

Options

Columns

Lists variables that are available for the chart.

Click Add another column to add more columns.

Group color

Turn on or off color groupings.

Summary

Select a statistical summary function (the method that is used for summarizing each category).

Two types of statistical summary functions are available. The distinction is important because it determines whether you need to specify a Value variable.

- **Functions that do not require a value variable.** Functions that do not require a variable. All count and percentage statistics are in this category. These statistics are available when the Value variable is not defined.
- **Functions that do require a value variable.** Functions that do require a Value variable. For example, the **Mean** function requires a variable on which the mean is calculated. These statistics are available when the Value variable is not defined.

Value

This field displays when a Summary function that requires a value variable, is selected. Select a variable to serve as the value.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Custom charts

The custom charts option provides options for pasting or editing JSON code to create the wanted chart.

Creating a custom chart

1. In the Chart Type section, click the Customized icon.

Customized



2. Paste the JSON code that contains the chart specifications into the provided JSON script field in the Details pane.

Dendrogram charts

Dendrogram charts are similar to tree charts and are typically used to illustrate a network structure (for example, a hierarchical structure). Dendrogram charts consist of a root node that is connected to subordinate nodes through edges or branches. The last nodes in the hierarchy are called leaves.

Creating a simple dendrogram chart

1. In the Chart Type section, click the Dendrogram icon.

Dendrogram



The canvas updates to display a Dendrogram chart template.

2. Select at least two variables from the Field name list. You can select SELECT ALL to select all available variables. A maximum number 300 variables are recommended.

Options

Field name

The list provides all available variables that represent the leave nodes. You must select at least two variables.

Show axis

The toggle control enables or disables the display of the distance value axis.

Linkage

The linkage criterion determines the distance between sets of observations as a function of the pairwise distances between observations.

Select Average distance, Min distance, or Max distance.

Tree layout

Left to right

The root node displays on the left and the leaf nodes display on the right.

Right to left

The root node displays on the right and the leaf nodes display on the left.

Top to bottom

The root node displays on the top and the leaf nodes display on the bottom.

Bottom to top

The root node displays on the bottom and the leaf nodes display on the top.

Radial

The root node displays in the middle and the leaf nodes radiate from the root.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Dual Y-axes charts

A dual Y-axes chart summarizes or plots two Y-axes variables that have different domains. For example, you can plot the number of cases on one axis and the mean salary on another. This chart can also be a mix of different graphic elements so that the dual Y-axes chart encompasses several of the different chart types. Dual Y-axes charts can display the counts as a line and the mean of each category as a bar.

Creating a simple Dual Y-axes chart

1. In the Chart Type section, click the Dual Y-axes icon.

Dual Y-axes



The canvas updates to display a Dual Y-axes chart template.

2. Select a variable as the X-axis variable.
 3. Select variable for the first Y-axis variable and then select a chart type to represent the variable (Bar, Line, or Scatter plot).
 4. Select variable for the second Y-axis variable and then select a chart type to represent the variable (Bar, Line, or Scatter plot).
- Note: You can use the up and down arrow controls to change the Y-axes order.

Options

X-axis

Lists variables that are available for the chart's X-axis.

Y-axis

Lists variables that are available for the chart's dual Y-axes.

Summary

When enabled, options for selecting the method that is used for summarizing each category are displayed.

Left Y-axis summary

Sets the summary method for the Y-axis that displays on the left side of the chart. Options include Sum, Mean, Maximum, and Minimum.

Right Y-axis summary

Sets the summary method for the Y-axis that displays on the right side of the chart. Options include Sum, Mean, Maximum, and Minimum.

Normalize data

When enabled, data is transformed into a normal distribution, which allows data from multiple data sets or columns to be easily compared.

Second axis lines

When enabled, the chart's second axis line is shown.

Reorder

When enabled, the chart's data is reordered based on the X and Y axis values.
Legend orient
Sets the chart legend orientation. Available options are Horizontal, Vertical, and Vertical bottom.
Primary title
The chart title.
Subtitle
The chart subtitle, which is placed directly beneath the chart title.
Footnote
The chart footnote, which is placed beneath the chart.
XAxis label
The x-axis label, which is placed beneath the x-axis.
YAxis label
The y-axis label, which is placed above the y-axis.

Error bar charts

Error bar charts represent the variability of data and indicate the error (or uncertainty) in a reported measurement. Error bars help determine whether differences are statistically significant. Error bars can also suggest goodness of fit for a specific function.

Creating a simple error bar chart

1. In the Chart Type section, click the Error bar icon.

Error bar



The canvas updates to display an error bar chart template.

2. Select a scale variable as the Category variable (the variable whose data is represented on the X-axis).
3. Select a variable as the Y-axis variable (the variable whose data is represented on the Y-axis).

Options

Category	Lists variables that are available for the chart's X-axis.
Y-axis	Lists variables that are available for the chart's Y-axis.
Category order	Select the order in which variable categories are sorted.
As read	Variable categories are presented as they appear in the data set.
Ascending	Sort variable categories in ascending order.
Descending	Sort variable categories in descending order.
Split by	Select a categorical variable that creates a table of charts, with a cell for each category in the Split by variable. Like grouping, split by variables essentially add more dimensions to your chart by displaying information for each variable category. For more information, see Adding Split by variables .
Reference line	When enabled, displays a reference line on the chart. The reference line correlates with the selected Statistical method.
Error bars	When enabled, the lines that represent the range of error are displayed in the chart.
Measure	Select the measure type that is represented by the error bars:
Confidence intervals	Sets the confidence intervals for the selected variables. The default value is 0.95 (95%), as reflected in the Represent value field.
Standard error	Measures the standard error of the selected variables.
Standard deviation	Measures the standard deviations of the selected variables.
Confidence level	This value represents the confidence intervals for the selected Measure. The default value is 0.95 (95%).

Statistical method

Select the method for describing the central tendency:

Mean

The result of summing the ratios and dividing the result by the total number ratios.

Median

The value such that number of ratios less than this value and the number of ratios greater than this value are the same.

Display mode

Select how the Statistical method selection displays (bar, line, or circle).

Legend orient

Sets the chart legend orientation. Available options are Horizontal, Vertical, and Vertical bottom.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

XAxis label

The x-axis label, which is placed beneath the x-axis.

YAxis label

The y-axis label, which is placed above the y-axis.

Evaluation charts

Evaluation charts are similar to histograms or collection graphs. Evaluation charts show how accurate models are in predicting particular outcomes. They work by sorting records based on the predicted value and confidence of the prediction, splitting the records into groups of equal size (quantiles), and then plotting the value of the criterion for each quantile, from highest to lowest. Multiple models are shown as separate lines in the plot.

Outcomes are handled by defining a specific value or range of values as a "hit". Hits usually indicate success of some sort (such as a sale to a customer) or an event of interest (such as a specific medical diagnosis).

Flag

Output fields are straightforward; hits correspond to `true` values.

Nominal

For nominal output fields, the first value in the set defines a hit.

Continuous

For continuous output fields, hits equal values greater than the midpoint of the field's range.

Evaluation charts can also be cumulative so that each point equals the value for the corresponding quantile plus all higher quantiles. Cumulative charts usually convey the overall performance of models better, whereas noncumulative charts often excel at indicating particular problem areas for models.

Creating a simple Evaluation chart

1. In the Chart Type section, click the Evaluation icon.

Evaluation



The canvas updates to display an Evaluation chart template.

2. Set the Target field, Predict field and Confidence field variables. The target field can be any instantiated flag or nominal field with two or more values. The predict field defines the variable that is used as the predicted value. The confidence field defines the variable that is used to establish the confidence of the prediction.

Note: The Predict field variable type must match the variable type that is selected for the Target field.

3. Specify a custom condition used to indicate the User defined hit. This option is useful for defining the outcome of interest rather than deducing it from the type of target field and the order of values.

You must specify a CLEM expression for a hit condition. For example, `@TARGET = "YES"` is a valid condition that indicates a value of `Yes` for the target field is counted as a hit in the evaluation. The specified condition is used for all target fields.

Options

Target field

Lists instantiated flag or nominal field variables with two or more values.

User defined hit

Specify a hit value. Hits indicate events of interest (for example, a specific medical diagnosis).

Predict field

Lists variables that can be used as the predicted value.

Confidence field

Lists variables that can establish the confidence of the prediction.

Cumulative plot

Create a cumulative chart when enabled. Values in cumulative charts are plotted for each quantile plus all higher quantiles.

Display mode

The settings control which charts display in preview mode and in the output.

Single mode

When selected, the Model Classification Tuning chart is in the only chart that displays in preview mode and in the output.

Classical mode

When selected, the Model Classification Tuning, Cutoff, Matrix Bar, ROC, Gains, ROI, and Profit charts display in preview mode and in the output.

Full mode

When selected, the Model Classification Tuning, Cutoff, Matrix Bar, ROC, Gains, ROI, Profit, GINI, Lift, and Response charts display in preview mode and in the output.

Evaluation charts

Cutoff

The cutoff chart shows the predicted versus actual values for selected variables for a specified cutoff value.

Matrix Bar

Matrix Bar charts are a good way to determine whether linear correlations exist between multiple variables.

ROC

ROC (Receiver Operating Characteristic) evaluates the performance of classification schemes where subjects are classified for one variable with two categories.

Gains

Gains are defined as the proportion of total hits that occurs in each quantile. Gains are computed as `(number of hits in quantile / total number of hits) × 100%`.

ROI

ROI (return on investment) is similar to profit in that it involves defining revenues and costs. ROI compares profits to costs for the quantile. ROI is computed as `(profits for quantile / costs for quantile) × 100%`.

Profit

Profit equals the revenue for each record minus the cost for the record. Profits for a quantile are the sum of profits for all records in the quantile. Revenues are assumed to apply only to hits, but costs apply to all records. Profits and costs can be fixed or can be defined by fields in the data. Profits are computed as `(sum of revenue for records in quantile – sum of costs for records in quantile)`.

Kolmogorov-Smirnov

Compares the observed cumulative distribution function for a variable with a specified theoretical distribution, which can be normal, uniform, exponential, or Poisson.

GINI

GINI measures statistical dispersion and is intended to represent the income or wealth distribution. It is the most commonly used measurement of inequality.

Lift

Lift compares the percentage of records in each quantile that are hits with the overall percentage of hits in the training data. It is computed as `(hits in quantile / records in quantile) / (total hits / total records)`.

Response

Response is the percentage of records in the quantile that are hits. Response is computed as `(hits in quantile / records in quantile) × 100%`.

Evaluation chart settings

The following settings apply only to profit and ROI charts.

Costs

Specify the fixed cost associated with each record.

Revenue

Specify the fixed revenue associated with each record that represents a hit.

Weight

If the records in your data represent more than one unit, you can use frequency weights to adjust the results. Specify the fixed weight associated with each record.

Heat map charts

Heat map charts present data where the individual values that are contained in a matrix are represented as colors.

Creating a simple heat map chart

1. In the Chart Type section, click the Heat map icon.

Heat map



The canvas updates to display a heat map chart template.

2. Select a variable as the Column variable. Each variable category is represented as an individual chart column.

3. Select a variable as the Row variable. Each variable category is represented as an individual chart row.

Options

Column

Lists variables that are available for the chart's columns. Each variable category is represented as an individual chart column.

Row

Lists variables that are available for the chart's rows. Each variable category is represented as an individual chart row.

Category order

Select the order in which variable categories are sorted.

As read

Variable categories are presented as they appear in the data set.

Ascending

Sort variable categories in ascending order.

Descending

Sort variable categories in descending order.

Summary

Select a statistical summary function for the graphic element. The result of the statistic determines the position of the graphic elements on the Y-axis. In a 2-D chart, the statistic is calculated for each value on the X-axis. In a 3-D chart, it is calculated for the intersection of values on the X-axis and Z-axis.

Two types of statistical summary functions are available. The distinction is important because it determines whether you need to specify a Value variable.

- **Functions that do not require a value variable.** Functions that do not require a variable. All count and percentage statistics are in this category. These statistics are available when the Value variable is not defined.
- **Functions that do require a value variable.** Functions that do require a Value variable. For example, the **Mean** function requires a variable on which the mean is calculated. These statistics are available when the Value variable is not defined.

Value

This field displays when a Summary function that requires a value variable, is selected. Select a variable to serve as the value.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

XAxis label

The x-axis label, which is placed beneath the x-axis.

YAxis label

The y-axis label, which is placed above the y-axis.

Histogram charts

A histogram is similar in appearance to a bar chart, but instead of comparing categories or looking for trends over time, each bar represents how data is distributed in a single category. Each bar represents a continuous range of data or the number of frequencies for a specific data point. Histograms are useful for showing the distribution of a single scale variable. Data are binned and summarized by using a count or percentage statistic. A variation of a histogram is a frequency polygon, which is like a typical histogram except that the area graphic element is used instead of the bar graphic element.

Another variation of the histogram is the population pyramid. Its name is derived from its most common use: summarizing population data. When used with population data, it is split by gender to provide two back-to-back, horizontal histograms of age data. In countries with a young population, the shape of the resulting graph resembles a pyramid.

Creating a histogram chart

1. In the Chart Type section, click the Histogram icon.

Histogram *



The canvas updates to display a histogram chart template.

2. Select a scale variable as the X-axis variable.

Note: The statistic for a histogram is Histogram or Histogram Percent. These statistics bin the data and calculate a count for each bin.

Options

X-axis

Lists variables that are available for the chart's X-axis.

Split by

Select a categorical variable that creates a table of charts, with a cell for each category in the Split by variable. Like grouping, split by variables essentially add more dimensions to your chart by displaying information for each variable category. For more information, see [Adding Split by variables](#).

Bin method

Specify a bin method that is used to create the chart bars. Available options include Auto bin, By bin width, and By bin num.

Show kde curve

When enabled, the kernel density estimate curve is shown on the chart.

Show distribution curve

When enabled, the distribution fitting curve is shown on the chart.

Distribution

The drop-down list provides the following distribution options.

Auto fit distribution

Automatically fits the distribution (the default setting).

Beta

Returns the value from a Beta distribution with specified shape parameters.

Exponential

Returns the value from an exponential distribution.

Gamma

Returns the value from the Gamma distribution, with the specified shape and scale parameters.

Log-normal

Returns the value from a log-normal distribution with specified parameters.

Normal

Returns the value from a normal distribution with specified mean and standard deviation.

Triangular

Returns the value from a triangular distribution with specified parameters.

Uniform

Returns the value from the uniform distribution between the minimum and maximum.

Weibull

Returns the value from a Weibull distribution with specified parameters.

Bin width

The slider controls the size of the interval that is used to split the data into groups.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

XAxis label

The x-axis label, which is placed beneath the x-axis.

YAxis label

The y-axis label, which is placed above the y-axis.

Line charts

A line chart plots a series of data points on a graph and connects them with lines. A line chart is useful for showing trend lines with subtle differences, or with data lines that cross one another. You can use a line chart to summarize categorical variables, in which case it is similar to a bar chart (see [Bar charts](#)). Line charts are also useful for time-series data.

Creating a simple time-series line chart

1. In the Chart Type section, click the Line icon.

Line •



The canvas updates to display a line chart template.

2. Select a date variable as the X-axis variable.

3. Select a scale variable as the Y-axis variable (the variable whose values were recorded over time).

Options

X-axis

Lists variables that are available for the chart's X-axis.

Y-axis

Lists variables that are available for the chart's Y-axis.

Split by

Select a categorical variable that creates a table of charts, with a cell for each category in the Split by variable. Like grouping, split by variables essentially add more dimensions to your chart by displaying information for each variable category. For more information, see [Adding Split by variables](#).

Area

When enabled, the area beneath the line is shown in a different color.

Smooth

When enabled, the chart shows a smooth curve.

Show data points

When enabled, the data point is shown in the chart.

Reorder

The toggle control reorders data based on X and Y-axis values.

Fit line

In a fit line, the data points are fitted to a line that usually does not pass through all of the data points. The fit line represents the trend of the data. Some fits lines are regression-based. Others are based on iterative weighted least squares. Select a fit line option from the drop-down list.

Show reference line

When enabled, the option shows a reference line on the chart that is based on the specified xAxiS and yAxiS values.

Enter a reference line value on xAxiS

When Show reference line is enabled, this setting provides the option of specifying a specific reference line value for the X-axis.
Click Add another column to specify more reference line values.

Enter a reference line value on yAxiS

When Show reference line is enabled, this setting provides the option of specifying a specific reference line value for the Y-axis.
Click Add another column to specify more reference line values.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

XAxis label

The x-axis label, which is placed beneath the x-axis.

YAxis label

The y-axis label, which is placed above the y-axis.

Map charts

Map charts are commonly used to compare values and show categories across geographical regions. Map charts are most beneficial when the data contains geographic information (countries, regions, states, counties, postal codes, and so on).

Creating a simple map chart

1. In the Chart Type section, click the Map icon.

Map



The canvas updates to display a map chart template.

2. Select a service to use for serving map images from the Map service drop-down list. The list provides options that cover specific global regions.
3. Select a map chart Type from the drop-down menu. The following options are available, dependent on the selected chart type:

Longitude

Select a variable to serve as the longitudinal value from the drop-down list.

Latitude

Select a variable to serve as the latitudinal value from the drop-down list.

Group

Select a variable that groups the data point locations from the drop-down menu.

Category

Select a column variable that you want to visualize.

Summary

Select a statistical summary function (the method that is used for summarizing each category).

Two types of statistical summary functions are available. The distinction is important because it determines whether you need to specify a Value variable.

- **Functions that do not require a value variable.** Functions that do not require a variable. All count and percentage statistics are in this category. These statistics are available when the Value variable is not defined.
- **Functions that do require a value variable.** Functions that do require a Value variable. For example, the **Mean** function requires a variable on which the mean is calculated. These statistics are available when the Value variable is not defined.

Value

This field displays when a Summary function that requires a value variable, is selected. Select a variable to serve as the value.

Options

Map service

Lists the services that are available for providing map images. See [Map service options](#).

Type

Lists the chart types that are available to represent the data.

Longitude

Lists the variables that are available to serve as the longitudinal value.

Latitude

Lists the variables that are available to serve as the latitudinal value.

Group

Lists the variables that can be used to group the data point locations.

Category

Lists column variables.

Summary

Select a statistical summary function (the method that is used for summarizing each category).

Two types of statistical summary functions are available. The distinction is important because it determines whether you need to specify a Value variable.

- **Functions that do not require a value variable.** Functions that do not require a variable. All count and percentage statistics are in this category. These statistics are available when the Value variable is not defined.
- **Functions that do require a value variable.** Functions that do require a Value variable. For example, the **Mean** function requires a variable on which the mean is calculated. These statistics are available when the Value variable is not defined.

Value

This field displays when a Summary function that requires a value variable, is selected. Select a variable to serve as the value.

Tooltip info

Lists the variables that can be used to generate tooltip information when the cursor hovers over a data point.

Size map

Lists available size map variables. These variables use differing sizes to represent themselves in the plot points.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

- [Map service options](#)
-

Map service options

On the Details pane for Map charts, you can select the map service to use for providing map images.

Configuring map service options

1. Stop the IBM® SPSS® Modeler Server.
2. If you're going to apply local geojson formatted map files, put them in the directory <Modeler_installation_directory>/dataview/conf/public/mapfiles. If the directory doesn't exist, create it.
3. Open the file <Modeler_installation_directory>/dataview/conf/application.conf in a text editor and add the following options to the `map.resources` section. In this example, two new maps named `World - Local Map` and `World - Online Map` are added.

```
map.resources {
    "mapservices" :
    [
        {
            "type"      : "geojson",
            "name"      : "World (Built-in)",
            "location"  : "built_in_world",
            "id"        : "built_in_world"
        },
        {
            "type"      : "geojson",
            "name"      : "Blank (Built-in)",
            "location"  : "built_in_blank",
            "id"        : "built_in_blank"
        },
        {
            "type"      : "geojson",
            "name"      : "World - Local Map",
            "location"  : "/mapfiles/world.json",
            "useProxy"   : "false",
            "id"        : "world_local"
        },
        {
            "type"      : "geojson",
            "name"      : "World - Online Map",
            "location"  : "http://map.example.com/map/world.json",
            "useProxy"   : "true",
            "id"        : "world_online"
        }
    ]
}
```

The following parameters are available for the map options:

Table 1. Map service parameters

Option	Value	Type	Required?	Description
<code>type</code>	<code>geojson</code>	String	Yes	Currently, only the value <code>geojson</code> is supported for this option.
<code>name</code>	Map name	String	Yes	Enter a name for the map. This name is shown in the Map service drop-down on the Details pane for map charts.
<code>location</code>	Map location	String	Yes	Enter the URL or local file location of the map. For URL, it must begin with <code>http://</code> or <code>https://</code> . For local file, it must begin with the path <code>/mapfiles/</code> .
<code>useProxy</code>	<code>true</code> or <code>false</code>	String	Yes	Use <code>true</code> if the map location is a URL. Use <code>false</code> if the map location is a local file.
<code>id</code>	Map ID	String	Yes	Use an unique ID for each map.

Note:

- Don't remove the default World (Built-in) and Blank (Built-in) options.
- Remember to use a comma between the content blocks as shown in the example, and be sure your new `map.resources` section is valid JSON formatted content.
- The IBM SPSS Modeler Server (or the standalone SPSS Modeler Client without Server) must be able to access the map location.

4. Start the IBM SPSS Modeler Server.

Now while working with map charts, on the Details pane, you'll see the new map service you added.

Math curve charts

A math curve chart plots mathematical equation curves that are based on user-entered expressions.

Creating a simple math curve chart

1. In the Chart Type section, click the Math curve icon.

Math curve



The canvas updates to display a math curve chart template.

2. Enter a starting value for the X-axis in the X value starts from field.
3. Enter an ending value for the X-axis in the X value ends with field.
4. Enter an equation that plots the graph curve in the equations field. Click Add another column to include more equations. Each equation treats **x** as an independent variable. The following equations are allowed.

- +, -, *, /, %, and ^
- abs(x)
- ceil(x)
- floor(x)
- log(x)
- max(a,b,c...)
- min(a,b,c...)
- random()
- round(x)
- sqrt(x)
- sin
- cos
- exp
- tan
- atan
- atan2
- asin
- acos

Options

X value starts from

The X-axis starting value.

X value ends with

The X-axis ending value.

equations

User-entered equations that plot the graph curve.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

XAxis label

The x-axis label, which is placed beneath the x-axis.

YAxis label

The y-axis label, which is placed above the y-axis.

Multi-chart charts

Multi-chart charts provide options for creating multiple charts. The charts can be of the same or different types, and can include different variables from the same data set.

Creating a simple multi-chart chart

1. In the Chart Type section, click the Multi-chart icon.

Multi-chart



The canvas updates to display a multi-chart chart template.

2. Click Add another sub-chart to add a subchart.
3. Select a chart type from the Type drop-down list.
4. Depending on the selected chart type, the following options are available:

Bar and Pie charts

Select a Category variable from the drop-down list.

Line and Scatter plot charts

Select an X-axis variable from the drop-down list.

Select an Y-axis variable from the drop-down list.

5. Click Add another sub-chart to include more charts types.

Options

Title

The subchart title.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Multiple series charts

Multiple series charts are similar to line charts, with the exception that you can chart multiple variables on the Y-axis.

Creating a simple multiple series chart

1. In the Chart Type section, click the Multi-series icon.

Multi-series



The canvas updates to display a multiple series chart template.

2. Select a variable as the X-axis variable.
3. Select at least two scale variables as the Y-axis variables.

Options

X-axis

Lists variables that are available for the chart's X-axis.

Y-axis

Lists variables that are available for the chart's Y-axes.

Select the chart type (bar, line, or scatter) from the drop-down list.

Click Add another column to include more columns on the chart.

Series style

Provides options for defining the Y-axes orientation. The following options are available.

- Default
- Separate Y axes
- Show a secondary Y axis
- Dual Y axes

Secondary Y-axis

Lists variables that are available for the chart's secondary Y-axes.

Select the chart type (bar, line, or scatter) from the drop-down list.

Click Add another column to include more columns on the chart.

Normalize data

When enabled, this setting transforms data into a normal distribution compares data from multiple data sets or multiple columns. This setting creates 100% stacking for counts and converts statistics to percents.

Reorder

When enabled, the chart's data is reordered based on the X and Y axis values.

Show label

When enabled, column labels display on the chart. Only scatter series data is supported.

Label field

The field menu provides variables to display as chart labels.

Legend orient

Sets the chart legend orientation. Available options are Horizontal, Vertical, and Vertical bottom.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

XAxis label

The x-axis label, which is placed beneath the x-axis.

YAxis label

The y-axis label, which is placed above the y-axis.

Parallel charts

Parallel charts are useful for visualizing high dimensional geometry and for analyzing multivariate data. Parallel charts resemble line charts for time-series data, but the axes do not correspond to points in time (a natural order is not present).

Creating a simple parallel chart

1. In the Chart Type section, click the Parallel icon.

Parallel



The canvas updates to display a parallel chart template.

2. Select at least two variables as the Columns variables. Each column represents a vertical, parallel axis in the chart.

Note: The column order is important for finding features. In a typical data analysis, you might need to reorder the columns numerous times.

Options

Columns

Lists variables that are available for the chart's Y-axes.

Click Add another column to add more columns.

Color map

Lists available color map variables. These variables use color progression, based on the range of values in the specified column, to represent themselves in the plot points. Color maps are also known as choropleth maps.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Pareto charts

Pareto charts contain both bars and a line graph. The bars represent individual variable categories and the line graph represents the cumulative total.

Creating a simple Pareto chart

1. In the Chart Type section, click the Pareto icon.

Pareto



The canvas updates to display a Pareto chart template.

2. Select a variable as the Category variable. The selected variable categories are drawn on the chart's X-axis.

Options

Category

Lists variables that are available for the chart's X-axis.

YAxis align

The toggle control enables and disables the alignment of the two Y-axes on the chart.

Gradient bar

The toggle control enables and disables the display of color gradients in the chart bars.

Highlight vital few

The toggle control enables and disables the highlighting of variable categories that are considered vital.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Pie charts

A pie chart is useful for comparing proportions. For example, you can use a pie chart to demonstrate that a greater proportion of Europeans is enrolled in a certain class.

Creating a simple pie chart

1. In the Chart Type section, click the Pie icon.

Pie •



The canvas updates to display a pie chart template.

2. Select a categorical (nominal or ordinal) variable from the Category list. The categories in this variable determine the number of slices in the pie chart.
3. Select a statistical summary function for the graphic element. For pie charts, you typically want a count-based statistic or a sum. The result of the statistic determines the size of each slice.

Options

Category

Select a categorical (nominal or ordinal) variable that determines the number of slices in the pie chart.

Summary

Select a statistical summary function for the graphic element. For pie charts, you typically want a count-based statistic or a sum. The result of the statistic determines the size of each slice.

Two types of statistical summary functions are available. The distinction is important because it determines whether you need to specify a Value variable.

- **Functions that do not require a value variable.** Functions that do not require a variable. All count and percentage statistics are in this category. These statistics are available when the Value variable is not defined.
- **Functions that do require a value variable.** Functions that do require a Value variable. For example, the **Mean** function requires a variable on which the mean is calculated. These statistics are available when the Value variable is not defined.

Value

This field displays when a Summary function that requires a scale variable, is selected. Select a variable to serve as the scale variable.

Pie type

The following styles are available.

Normal

The pie segments display as normal slices.

Ring

The pie segments display as a ring. This style is also known as a doughnut chart.

Rose

Unlike the normal pie chart, which uses a common radius, the pie segment sizes vary depending on their value.

Rose area

Unlike the normal pie chart, which uses a common radius, the pie segment sizes vary depending on their area.

Rose ring

Unlike the normal pie chart, which uses a common radius, the pie segment sizes vary depending on their value and the segments display as a ring.

Half rose

Same as Rose, except the chart is represented as one-half of a pie.

Show value

When enabled, the pie slice values display in the legend.

Outer field

The list provides variables that can be used as the count value of the outer ring.

Outer label angle

The value specifies the location of the outer field variable value on the outer ring.

Legend orient

Sets the chart legend orientation. Available options are Horizontal, Vertical, and Vertical bottom.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Population pyramid charts

Population pyramid charts (also known as "age-sex pyramids") are commonly used to present and analyze population information based on age and gender.

Creating a simple Population pyramid chart

1. In the Chart Type section, click the Population pyramid icon.

Population ...



The canvas updates to display a Population pyramid chart template.

2. Select a Y-axis variable from the drop-down list.
3. Select a Split by variable from the drop-down list.

Options

Y-axis

Lists variables that are available for the chart's Y-axis.

Split by

Select a categorical variable that creates a table of charts, with a cell for each category in the Split by variable. Like grouping, split by variables essentially add more dimensions to your chart by displaying information for each variable category. For more information, see [Adding Split by variables](#).

Bin width

The slider controls the size of the interval that is used to split the data into groups.

Show distribution curve

When enabled, the distribution fitting curve is shown on the chart.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Q-Q plots

Q-Q (quantile-quantile) plots compare two probability distributions by plotting their quantiles against each other. A Q-Q plot is used to compare the shapes of distributions, providing a graphical view of how properties such as location, scale, and skewness are similar or different in the two distributions.

Creating a simple Q-Q plot chart

1. In the Chart Type section, click the Q-Q plot icon.

Q-Q plot



The canvas updates to display a Q-Q plot chart template.

2. Select a variable as the X-axis variable.

Options

X-axis

Lists variables that are available for the chart's X-axis.

Distribution

The drop-down list provides all available distribution methods.

Auto fit distribution

Automatically fits the distribution (the default setting).

Beta

Returns the value from a Beta distribution with specified shape parameters.

Exponential

Returns the value from an exponential distribution.

Gamma

Returns the value from the Gamma distribution, with the specified shape and scale parameters.

Log-normal

Returns the value from a log-normal distribution with specified parameters.

Normal

Returns the value from a normal distribution with specified mean and standard deviation.

Uniform

Returns the value from the uniform distribution between the minimum and maximum.

Student t

Returns the value from Student's t distribution, with specified degrees of freedom.

Plot type

Select either a Q-Q (quantile-quantile) plot or a P-P (percent-percent) plot.

Auto fit

When enabled, data parameters for the selected Distribution are automatically estimated. When not selected, the Shape and Scale distribution values display.

Shape1

Sets the shape1 value for the Beta distribution. This setting is available only when Auto fit is not enabled and Beta is selected as the Distribution.

Shape2

Sets the shape2 value for the Beta distribution. This setting is available only when Auto fit is not enabled and Beta is selected as the Distribution.

Shape

Sets the shape value for the selected distribution. This setting is available only when Auto fit is not enabled and Gamma or Log-normal is selected as the Distribution.

Scale

Sets the scale value for the selected distribution. This setting is available only when Auto fit is not enabled and Exponential, Gamma, or Log-normal is selected as the Distribution.

Mean

Sets the means value for the Normal distribution. This setting is available only when Auto fit is not enabled and Normal is selected as the Distribution.

Std. dev.

Sets the standard deviation value for the Normal distribution. This setting is available only when Normal is selected as the Distribution.

Min

Sets the minimum value for the Uniform distribution. This setting is available only when Auto fit is not enabled and Uniform is selected as the Distribution.

Max

Sets the maximum value for the Uniform distribution. This setting is available only when Auto fit is not enabled and Uniform is selected as the Distribution.

Degrees of freedom(df)

Set the degrees of freedom value for the Student t distribution. This setting is available only when Auto fit is not enabled and Student t is selected as the Distribution.

Detrend

When enabled, detrended plots display on the chart.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

- The chart footnote, which is placed beneath the chart.
- XAxis label
 - The x-axis label, which is placed beneath the x-axis.
- YAxis label
 - The y-axis label, which is placed above the y-axis.

Radar charts

Radar charts compare multiple quantitative variables and are useful for visualizing which variables have similar values, or if outliers exist among the variables. Radar charts consists of a sequence of spokes, with each spoke representing a single variable. Radar Charts are also useful for determining which variables are scoring high or low within a data set.

Creating a simple radar chart

- In the Chart Type section, click the Radar icon.

Radar



The canvas updates to display a radar chart template.

- Select a Columns variable from the drop-down list.

Note: Click Add another column to include more columns. At least three columns variables must be defined.

Options

Category

Select a categorical (nominal or ordinal) variable. If you select None as the category values, all values are shown separately, and no summary method is applied.

Summary

When a categorical variable is selected, the following summary statistics are available.

Count

Total number of cases.

Sum

Sum of the values.

Mean

Arithmetic average; the sum divided by the number of cases.

Maximum

Largest (highest) value.

Minimum

Smallest (lowest) value.

Radar Layout

Determines the background image layout for the radar chart:

Circle

When selected, the radar chart is drawn over a circular layout.

Polygon

When selected, the radar chart is drawn over a polygonal layout.

Split by

Select a categorical variable that creates a table of charts, with a cell for each category in the Split by variable. Like grouping, split by variables essentially add more dimensions to your chart by displaying information for each variable category. For more information, see [Adding Split by variables](#).

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Relationship charts

A relationship chart is useful for determining how variables relate to each other.

Creating a simple relationship chart

1. In the Chart Type section, click the Relationship icon.

Relationship



The canvas updates to display a relational chart template.

2. Select at least two variables as Columns variables.

Note: Click Add another column to include more column variables.

Options

Columns

Lists the available variables.

Click Add another column to add more columns.

Line style

Controls the line style between related data points.

Curve

When selected, curved lines are drawn between related data points.

Straight

When selected, straight lines are drawn between related data points.

Label threshold

Displays labels for data points whose values exceed the defined value.

Legend orient

Sets the chart legend orientation. Available options are Horizontal, Vertical, and Vertical bottom.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Scatter plots and dot plots

Several broad categories of charts are created with the point graphic element.

Scatter plots

Scatter plots are useful for plotting multivariate data. They can help you determine potential relationships among scale variables. A simple scatter plot uses a 2-D coordinate system to plot two variables. A 3-D scatter plot uses a 3-D coordinate system to plot three variables. When you need to plot more variables, you can try overlay scatter plots and scatter plot matrices (SPLOMs). An overlay scatter plot displays overlaid pairs of X-Y variables, with each pair distinguished by color or shape. A SPLOM creates a matrix of 2-D scatter plots, with each variable plotted against every other variable in the SPLOM.

Dot plots

Like histograms, dot plots are useful for showing the distribution of a single scale variable. The data are binned, but, instead of one value for each bin (like a count), all of the points in each bin are displayed and stacked. These graphs are sometimes called density plots.

Summary point plots

Summary point plots are similar to bar charts, except that points are drawn in place of the top of the bars. For more information, see [Bar charts](#).

Drop-line charts

Drop-line charts are a special type of summary point plot. The points are grouped and a line is drawn through the points in each category. The drop-line chart is useful for comparing a statistic across categorical variables.

Creating a simple scatter plot

1. In the Chart Type section, click the Scatter plot icon.

Scatter plot



The canvas updates to display a scatter plot chart template.

2. Select a scale variable as the X-axis variable.
3. Select a scale variable as the Y-axis variable. You do not need to specify a statistic because scatter plots typically show raw values.

Options

- X-axis
Lists variables that are available for the chart's X-axis.
- Y-axis
Lists variables that are available for the chart's Y-axis.
- Color map
Lists available color map variables. These variables use color progression, based on the range of values in the specified column, to represent themselves in the plot points. Color maps are also known as choropleth maps.
- Size map
Lists available size map variables. These variables use differing sizes to represent themselves in the plot points.
- Shape map
Lists available shape map variables. These variables use differing shapes to represent themselves in the plot points.
- Fit line
In a fit line, the data points are fitted to a line that usually does not pass through all of the data points. The fit line represents the trend of the data. Some fits lines are regression-based. Others are based on iterative weighted least squares. Select a fit line option from the drop-down list.
- Gradient bubble
The toggle control enables and disables the display of color gradients and 3D effects in the chart bubbles. The setting is not available when a Color map variable is selected.
- Minimum bubble size
Sets the minimum bubble size. Enter a value in the range 5 - 20.
- Maximum bubble size
Sets the maximum bubble size. Enter a value in the range 20 - 80.
- Show reference line
When enabled, the option shows a reference line on the chart that is based on the specified xAixs and yAixs values.
- Enter a reference line value on xAixs
When Show reference line is enabled, this setting provides the option of specifying a specific reference line value for the X-axis.
Click Add another column to specify more reference line values.
- Enter a reference line value on yAixs
When Show reference line is enabled, this setting provides the option of specifying a specific reference line value for the Y-axis.
Click Add another column to specify more reference line values.
- Show label
When enabled, column labels display on the chart. Only scatter series data is supported.
- Label field
The field menu provides variables to display as chart labels.
- Primary title
The chart title.
- Subtitle
The chart subtitle, which is placed directly beneath the chart title.
- Footnote
The chart footnote, which is placed beneath the chart.
- XAxis label
The x-axis label, which is placed beneath the x-axis.
- YAxis label
The y-axis label, which is placed above the y-axis.

Scatter matrix charts

Scatter plot matrices are a good way to determine whether linear correlations exist between multiple variables.

Creating a scatter matrix chart

1. In the Chart Type section, click the Scatter matrix icon.

Scatterplot matrix



The canvas updates to display a scatter matrix chart template.

2. Select at least two scale Columns variables.

Note: Click Add another column to include more column variables.

Each selected variable is plotted against every other variable to create a matrix of individual scatter plots.

Options

Columns

Select at least two matrix variables. The variables must be numeric (but not date format).

Click Add another column to add more columns.

Color map

Lists available color map variables. These variables use color progression, based on the range of values in the specified column, to represent themselves in the plot points. Color maps are also known as choropleth maps.

Correlation

When enabled, linear correlation information (Strong, Medium, Weak) is shown for the selected variables.

Show kde curve

When enabled, the kernel density estimate curve is shown on the chart.

Show histogram

When enabled, histogram charts display for the selected column variables.

Gradient bubble

The toggle control enables and disables the display of color gradients and 3D effects in the chart bubbles. The setting is not available when a Color map variable is selected.

Minimum bubble size

Sets the minimum bubble size. Enter a value in the range 5 - 20.

Maximum bubble size

Sets the maximum bubble size. Enter a value in the range 20 - 80.

Show label

When enabled, column labels display on the chart. Only scatter series data is supported.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Series array charts

Series array charts include individual sub charts and display the Y-axis for all sub charts in the legend.

Creating a series array chart

1. In the Chart Type section, click the Series array icon.

Series array



The canvas updates to display a series array chart template.

2. Click Add another sub-chart to define the first series array sub chart.
3. Select a variable as the X-axis variable.
4. Select a variable as the Y-axis variable.
5. Click Add another sub-chart to define more series array sub charts.

Options

X-axis

Lists variables that are available for the sub chart's x-axis.

Y-axis

Lists variables that are available for the sub chart's y-axis.

Split by

Select a categorical variable that creates a table of charts, with a cell for each category in the Split by variable. Like grouping, split by variables essentially add more dimensions to your chart by displaying information for each variable category. For more information, see [Adding Split by variables](#).

Legend label

	Specify a sub-chart legend title.
Title	Specify a sub-chart title.
Type	Select a sub-chart type. Available options are line, bar, and scatter.
Smooth	When enabled, the chart shows a smooth curve.
Show data points	When enabled, the data point is shown in the chart.
Primary title	The chart title.
Subtitle	The chart subtitle, which is placed directly beneath the chart title.
Footnote	The chart footnote, which is placed beneath the chart.
XAxis label	The x-axis label, which is placed beneath the x-axis.
YAxis label	The y-axis label, which is placed above the y-axis.

Sunburst charts

A sunburst chart is useful for visualizing hierarchical data structures. A sunburst chart consists of an inner circle that is surrounded by rings of deeper hierarchy levels. The angle of each segment proportional to either a value or divided equally under its inner segment. The chart segments are colored based on the category or hierarchical level to which they belong.

Creating a simple sunburst chart

1. In the Chart Type section, click the Sunburst icon.

Sunburst



The canvas updates to display a sunburst chart template.

2. Select a categorical (nominal or ordinal) variable from the Columns list. The categories in this variable determine the number of segments in the chart.
3. Click Add another column and select another categorical (nominal or ordinal) variable from the Columns list. The categories in this variable determine the number of segments in the chart's second ring and represent a hierarchical level.
4. Select a statistical summary function for the graphic element (count-based statistic or a sum). The result of the statistic determines the size of each segment. When Sum is selected, choose a scale variable from the Value list to represent the value in the data set to summarize.
5. Select a Sunburst layout option (either Traditional or Divergent).

Options

Columns

Select a categorical (nominal or ordinal) variable that determines the number of segments in the chart.

Summary

Select a statistical summary function for the graphic element (count-based statistic or a sum). The result of the statistic determines the size of each slice.

Two types of statistical summary functions are available. The distinction is important because it determines whether you need to specify a Value variable.

Functions that do not require a value variable

All count and percentage statistics are in this category. These statistics are available when there is no defined Value variable.

Functions that do require a value variable

For example, the **Sum** function requires a variable on which the summary is calculated.

Value

This field displays when a Summary function that requires a scale variable, is selected. Select a variable to serve as the scale variable.

Sunburst layout

Available options are Traditional and Divergent.

Primary title

- The chart title.
Subtitle
The chart subtitle, which is placed directly beneath the chart title.
Footnote
The chart footnote, which is placed beneath the chart.
-

t-SNE charts

T-distributed Stochastic Neighbor Embedding (t-SNE) is a machine learning algorithm for visualization. t-SNE charts model each high-dimensional object by a two-or-three dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability.

Creating a simple t-SNE chart

1. In the Chart Type section, click the t-SNE icon.

The canvas updates to display a t-SNE chart template.
2. Set the Perplexity, Learning rate, and Maximum iterations values.
3. Optional: Select a Color map variable.

Options

- Perplexity
Sets a number that establishes an educated guess as to the number of close neighbors for each data point. The purpose is to balance the local and global aspects for your data.
- Learning rate
This value affects the speed of learning by specifying the weight size changes at each iteration.
- Maximum iterations
The maximum number of iterations to run.
- Color map
Lists available color map variables. These variables use color progression, based on the range of values in the specified column, to represent themselves in the plot points. Color maps are also known as choropleth maps.
- Primary title
The chart title.
- Subtitle
The chart subtitle, which is placed directly beneath the chart title.
- Footnote
The chart footnote, which is placed beneath the chart.
-

Time plots

Time plots illustrate data points at successive intervals of time. The time series you plot must contain numeric values and are assumed to occur over a range of time in which the periods are uniform. Time plots provide a preliminary analysis of the characteristics of time series data on basic statistics and test, and thus generate useful insights about your data before modeling. Time plots include analysis methods such as decomposition, augmented Dickey-Fuller test (ADF), correlations (ACF/PACF), and spectral analysis.

Creating a simple time plot

1. In the Chart Type section, click the Time plot icon.

The canvas updates to display a time plot chart template.
2. From the Values drop-down, select a value for the y-axis.

Options

Values

Lists variables that are available for the time plot values.

Date

Select a date from the drop-down, if applicable. Each observation is separated by the same time interval. If you select a date variable, a resample option is shown. You can use this option to aggregate value fields to match the specified interval.

Time plot algorithm

The time plot algorithm to use for analyzing the time series data.

Decomposition

Decompose a time series as three components (trend-cycle, seasonal, and irregular). Decomposition is run in an additive fashion.

ADF test

Augmented Dickey–Fuller (ADF) tests the null hypothesis that a unit root exists in the series and the series is not stationary. If the test result rejects the null hypothesis, the series is stationary, or can be represented to be stationary with a difference model.

ACF/PACF

Correlations of series.

Spectral analysis

An analytic tool in the frequency domain. The highest frequency is marked as diamond.

Swap chart position

Reverses the positions of the charts on the planimetric rectangular coordinate system and the polar coordinate system.

Show the turning point

Shows or hides the turning points in the charts. The series is explored to determine whether it has an overall trend or has some turning points that change the direction of the trend pattern.

Show the outlier

Shows or hides any outliers. The outliers of the time series are analyzed from the irregular component of time series decomposition.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

XAxis label

The x-axis label, which is placed beneath the x-axis.

YAxis label

The y-axis label, which is placed above the y-axis.

Theme River charts

A theme river is a specialized flow graph that shows changes over time.

Creating a simple theme river chart

1. In the Chart Type section, click the Theme River icon.

Theme River



The canvas updates to display a theme river chart template.

2. Select an X-axis variable.
3. Select a Category variable.

Restriction: If a category field has more than 50 distinct categories, only the first 50 maximum categories are used as events in the chart.

Options

X-axis

Lists variables that are available for the X-axis.

Order based on

Depending on the X-axis value you select, you might be able to specify whether the category order is based on category name or category value.

Category order

Depending on the X-axis value you select, you might be able to specify whether the category order is ascending, descending, or as read from the data set.

Category

Lists variables that are available as categories.

Summary

Select a statistical summary function (the method that is used for summarizing each category).

Two types of statistical summary functions are available. The distinction is important because it determines whether you need to specify a Value variable.

- **Functions that do not require a value variable.** Functions that do not require a variable. All count and percentage statistics are in this category. These statistics are available when the Value variable is not defined.
- **Functions that do require a value variable.** Functions that do require a Value variable. For example, the **Mean** function requires a variable on which the mean is calculated. These statistics are available when the Value variable is not defined.

Value

This field displays when a Summary function that requires a value variable, is selected. Select a variable to serve as the value.

Legend orient

Sets the chart legend orientation. Available options are Horizontal, Vertical, and Vertical bottom.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Tree charts

Tree charts represent hierarchy in a tree-like structure. The structure of a Tree chart consists of a root node (has no parent node), line connections (named branches), and leaf nodes (have no child nodes). Line connections represent the relationships and connections between the members.

Creating a simple tree chart

1. In the Chart Type section, click the Tree icon.



The canvas updates to display a Tree chart template.

2. Select a Columns variable from the drop-down list.

Note: Click Add another column to include more column variables.

Options

Columns

Lists variables that are available to represent chart columns.

Summary

Select a statistical summary function for the graphic element. The result of the statistic determines the position of the graphic elements.

Two types of statistical summary functions are available. The distinction is important because it determines whether you need to specify a Value variable.

Functions that do not require a value variable

All count and percentage statistics are in this category. These statistics are available when there is no defined Value variable.

Functions that do require a value variable

For example, the **Mean** function requires a variable on which the mean is calculated. These statistics are available when there is a defined Value variable.

Value

This field displays when a Summary function that requires a value variable, is selected. Select a variable to serve as the basis for the summary value.

Tree layout

Left to right

The root node displays on the left and the leaf nodes display on the right.

Right to left

The root node displays on the right and the leaf nodes display on the left.

Top to bottom

The root node displays on the top and the leaf nodes display on the bottom.

Bottom to top

The root node displays on the bottom and the leaf nodes display on the top.

Radial

The root node displays in the middle and the leaf nodes radiate from the root.

Leaf depth

Sets the drill-down level value for the leaf nodes.

Show leaves label

When enabled, labels display for each leaf node.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Treemap charts

Treemap charts are an alternative method for visualizing the hierarchical structure of tree diagrams while also displaying quantities for each category. Treemap charts are useful for identifying patterns in data. Tree branches are represented by rectangles, with each sub branch represented by smaller rectangles.

Creating a simple Treemap chart

1. In the Chart Type section, click the Treemap icon.

Treemap



The canvas updates to display a Treemap chart template.

2. Select a Columns variable from the drop-down list.

Note: Click Add another column to include more column variables.

Options

Columns

Lists variables that are available to represent chart columns.

Summary

Select a statistical summary function for the graphic element. The result of the statistic determines the position of the graphic elements.

Two types of statistical summary functions are available. The distinction is important because it determines whether you need to specify a Value variable.

Functions that do not require a value variable

All count and percentage statistics are in this category. These statistics are available when there is no defined Value variable.

Functions that do require a value variable

For example, the **Mean** function requires a variable on which the mean is calculated. These statistics are available when there is a defined Value variable.

Value

This field displays when a Summary function that requires a value variable is selected. Select a variable to serve as the basis for the summary value.

Leaf depth

Sets the drill-down level value for the leaf nodes.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Word cloud charts

Word cloud charts present data as words, where the size and placement of any individual word is determined by how it is weighted.

Creating a simple word cloud chart

1. In the Chart Type section, click the Word cloud icon.

Word cloud



The canvas updates to display a word cloud chart template.

2. Select a variable as the Source variable. Each variable category is represented in the chart based on its weight value.
3. Select a Shape value for the chart. The chart data is presented in the selected shape.

Options

Source

Lists variables that are available as the chart's source. Each variable category is represented in the chart based on its weight value.

Shape

Lists the available chart shapes. The chart data is presented in the selected shape.

Primary title

The chart title.

Subtitle

The chart subtitle, which is placed directly beneath the chart title.

Footnote

The chart footnote, which is placed beneath the chart.

Global visualization preferences

You can override the default settings for titles, range slider, grid lines, and mouse tracking. You can also specify a different color scheme template.

1. After right-clicking a data node and selecting View Data, click the Global visualization preferences control in the Actions section.

Figure 1. Global visualization preferences control



The Global visualization preferences dialog provides the following settings.

Titles

Provides global chart title settings.

Global titles

Enables or disables the global titles for all charts.

Global primary title

Enables or disables the display of global, primary chart titles. When enabled, the top-level chart title that you enter here is applied to all chart's, effectively overriding each chart's individual Primary title setting.

Global subtitle

Enables or disables the display of global chart subtitles. When enabled, the chart subtitle that you enter here is applied to all chart's, effectively overriding each chart's individual Subtitle setting.

Default titles

Enables or disables the default titles for all charts.

Title alignment

Provides the title alignment options Left, Center (the default setting), and Right.

Tools

Provides options that control chart behavior.

Range slider

Enables or disables the range slider for each chart. When enabled, you can control the amount of chart data that displays with a range slider that is provided for each chart.

Grid lines

Controls the display of X axis (vertical) and Y axis (horizontal) grid lines.

Mouse tracker

When enabled, the mouse cursor location, in relation to the chart data, is tracked and displayed when placed anywhere over the chart.

Toolbox

Enables or disables the toolbox for each chart. Depending on the chart type, the toolbox on the right of the screen provides tools such as zoom, save as image, restore, select data, and clear selection.

ARIA

When enabled, web content and web applications are more accessible to users with disabilities.

Filter out null

Enables or disables the filtering of null chart data.

X axis on zero

When enabled, the X axis lies on the other's origin position. When not enabled, the X axis always starts at 0.

Y axis on zero

When enabled, the Y axis lies on the other's origin position. When not enabled, the Y axis always starts at 0.

Show xAxis Label

Enables or disables the xAxis label.

Show yAxis Label

Enables or disables the yAxis label.

Show xAxis Line

Enables or disables the xAxis line.

Show yAxis Line

Enables or disables the yAxis line.

Show xAxis Name

Enables or disables the xAxis name.

Show yAxis Name

Enables or disables the yAxis name.

yAxis Name Location

The drop-down list provides options for specifying the yAxis name location. Options include Start, Middle, and End.

Truncation length

The specified value sets the string length. Strings that are longer than the specified length are truncated. The default value is 10. When 0 is specified, truncation is turned off.

xAxis tick label decimal

Sets the tick label decimal value for the xAxis. The default value is 3.

yAxis tick label decimal

Sets the tick label decimal value for the yAxis. The default value is 3.

xAxis tick label rotate

Sets the xAxis tick label rotation value. The default value is 0 (no rotation). You can specify value in the range -90 to 90 degrees.

Theme

Select a template to change the colors that are used in charts that have a grouping or stacking variable. Any element attributes defined in the selected template file override the default template settings for those element attributes.

2. Click Apply to save your settings or Cancel to disregard the changes.

Dashboard

You can create a chart dashboard layout for viewing several charts at a time. Save layouts as templates and drag-and-drop your saved charts to the positions in your layout.

1. After right-clicking a data node and selecting View Data, click the Dashboard control in the Actions section.

Figure 1. Dashboard control



The Dashboard displays and provides the following settings.

Template

In this section, you can create new layout templates or choose from predefined templates or from your saved templates.

Choose a layout template

Choose from available layout templates or start with a new template.

Actions

Click the Edit the layout icon to edit the layout template you selected. You can also import a dashboard file. When finished, save your template.

Click the Leave layout edit mode icon when you're finished.

Content

Use this section to drag-and-drop items to your dashboard layout.

Choose a saved chart

This section provides a list of your saved charts. Drag-and-drop charts to the desired position on your dashboard layout.

Choose an object

You can also drag HTML text or images to your dashboard layout.

Working with output

When you run some streams, the results are available in the Viewer via the Model tab or the Advanced tab of model nugget nodes. In the Viewer, you can easily navigate to the output that you want to see. You can also manipulate the output and create a document that contains precisely the output that you want. Some graph output also uses the Viewer.

The Viewer is used for the following output in IBM® SPSS® Modeler:

- TCM model nuggets
 - STP model nuggets
 - TwoStep-AS Cluster model nuggets
 - GSAR model nuggets
 - Map Visualization graph node
- [Viewer](#)
 - [Copying output into other applications](#)
 - [Interactive output](#)
 - [Export output](#)
 - [Viewer printing](#)
 - [Saving output](#)
 - [Pivot tables](#)
 - [Options](#)
-

Viewer

Results are displayed in the Viewer. You can use the Viewer to:

- Browse results
- Show or hide selected tables and charts
- Change the display order of results by moving selected items
- Move items between the Viewer and other applications

The Viewer is divided into two panes:

- The left pane contains an outline view of the contents.
- The right pane contains statistical tables, charts, and text output.

You can click an item in the outline to go directly to the corresponding table or chart. You can click and drag the right border of the outline pane to change the width of the outline pane.

- [Showing and hiding results](#)
 - [Moving, deleting, and copying output](#)
 - [Changing initial alignment](#)
 - [Changing alignment of output items](#)
 - [Viewer outline](#)
 - [Editing and adding items to the Viewer](#)
 - [Finding and replacing information in the Viewer](#)
-

Showing and hiding results

In the Viewer, you can selectively show and hide individual tables or results from an entire procedure. This process is useful when you want to shorten the amount of visible output in the contents pane.

- [To hide tables and charts](#)
- [To hide procedure results](#)

Related information

- [Controlling the initial display state for new output](#)
 - [Viewer](#)
-

To hide tables and charts

1. Double-click the item's book icon in the outline pane of the Viewer.
or
2. Click the item to select it.
3. From the menus choose:
View > Hide
4. Click the closed book (Hide) icon on the Outlining toolbar.

The open book (Show) icon becomes the active icon, indicating that the item is now hidden.

Related information

- [Showing and hiding results](#)
 - [To hide procedure results](#)
 - [Controlling the initial display state for new output](#)
-

To hide procedure results

1. Click the box to the left of the procedure name in the outline pane.

This hides all results from the procedure and collapses the outline view.

Related information

- [Showing and hiding results](#)
 - [To hide tables and charts](#)
 - [Controlling the initial display state for new output](#)
-

Moving, deleting, and copying output

You can rearrange the results by copying, moving, or deleting an item or a group of items.

- [Moving output in the Viewer](#)
 - [Deleting output in the Viewer](#)
-

Moving output in the Viewer

1. Select the items in the outline or contents pane.
2. Drag and drop the selected items into a different location.

The item you cut in step 1, is moved to the selected location.

Related information

- [Moving, deleting, and copying output](#)
 - [Deleting output in the Viewer](#)
-

Deleting output in the Viewer

1. Select the items in the outline or contents pane.
2. Press the Delete key.
or
3. From the menus choose:
`Edit > Delete`

Related information

- [Moving, deleting, and copying output](#)
 - [Moving output in the Viewer](#)
-

Changing initial alignment

By default, all results are initially left-aligned. To change the initial alignment of new output items:

1. From the menus choose:
`Edit > Options`
 2. Click the Viewer tab.
 3. In the Initial Output State group, select the item type (for example, pivot table, chart, text output).
 4. Select the alignment option you want.
-

Changing alignment of output items

1. In the outline or contents pane, select the items that you want to align.
2. From the menus choose:
`Format > Align Left`
or
`Format > Center`
or
`Format > Align Right`

Viewer outline

The outline pane provides a table of contents of the Viewer document. You can use the outline pane to navigate through your results and control the display. Most actions in the outline pane have a corresponding effect on the contents pane.

- Selecting an item in the outline pane displays the corresponding item in the contents pane.
- Moving an item in the outline pane moves the corresponding item in the contents pane.
- Collapsing the outline view hides the results from all items in the collapsed levels.

Controlling the outline display

To control the outline display, you can:

- Expand and collapse the outline view
- Change the outline level for selected items
- Change the size of items in the outline display
- Change the font that is used in the outline display
- [Collapsing and expanding the outline view](#)
- [Changing the outline level](#)

- [To change the size of outline items](#)
 - [To change the font in the outline](#)
-

Collapsing and expanding the outline view

1. Click the box to the left of the outline item that you want to collapse or expand.
or
 2. Click the item in the outline.
 3. From the menus choose:
`View > Collapse`
or
`View > Expand`
-

Changing the outline level

1. Click the item in the outline pane.
2. From the menus choose:
`Edit > Outline > Promote`
or
`Edit > Outline > Demote`

Related information

- [Collapsing and expanding the outline view](#)
-

To change the size of outline items

1. From the menus choose:
`View > Outline Size`
2. Select the outline size (Small, Medium, or Large).

Related information

- [Viewer outline](#)
 - [To change the font in the outline](#)
-

To change the font in the outline

1. From the menus choose:
`View > Outline Font...`
2. Select a font.

Related information

- [Viewer outline](#)
 - [To change the size of outline items](#)
-

Editing and adding items to the Viewer

In the Viewer, you can edit and add items such as titles, new text, charts, or material from other applications.

- [Adding a title or text](#)
 - [To add a text file](#)
 - [Pasting objects into the Viewer](#)
-

Adding a title or text

Text items that are not connected to a table or chart can be added to the Viewer.

1. Click the table, chart, or other object that will precede the title or text.
2. From the menus choose:
`Insert > New Title`
or
`Insert > New Text`
3. Double-click the new object.
4. Enter the text.

Related information

- [To add a text file](#)
-

To add a text file

1. In the outline pane or contents pane of the Viewer, click the table, chart, or other object that will precede the text.
2. From the menus choose:
`Insert > Text File...`
3. Select a text file.

To edit the text, double-click it.

Related information

- [Adding a title or text](#)
-

Pasting objects into the Viewer

Objects from other applications can be pasted into the Viewer. You can use either Paste After or Paste Special. Either type of pasting puts the new object after the currently selected object in the Viewer. Use Paste Special when you want to choose the format of the pasted object.

Finding and replacing information in the Viewer

1. To find or replace information in the Viewer, from the menus choose:
`Edit > Find`
or
`Edit > Replace`

You can use Find and Replace to:

- Search the entire document or just the selected items.
- Search down or up from the current location.
- Search both panes or restrict the search to the contents or outline pane.

- Search for hidden items. These include any items hidden in the contents pane (for example, Notes tables, which are hidden by default) and hidden rows and columns in pivot tables.
- Restrict the search criteria to case-sensitive matches.
- Restrict the search criteria in pivot tables to matches of the entire cell contents.
- Restrict the search criteria in pivot tables to footnote markers only. This option is not available if the selection in the Viewer includes anything other than pivot tables.

Hidden Items and Pivot Table Layers

- Layers beneath the currently visible layer of a multidimensional pivot table are not considered hidden and will be included in the search area even when hidden items are not included in the search.
- Hidden items include hidden items in the contents pane (items with closed book icons in the outline pane or included within collapsed blocks of the outline pane) and rows and columns in pivot tables either hidden by default (for example, empty rows and columns are hidden by default) or manually hidden by editing the table and selectively hiding specific rows or columns. Hidden items are only included in the search if you explicitly select **Include hidden items**.
- In both cases, the hidden or nonvisible element that contains the search text or value is displayed when it is found, but the item is returned to its original state afterward.

Finding a range of values in pivot tables

To find values that fall within a specified range of values in pivot tables:

1. Activate a pivot table or select one or more pivot tables in the Viewer. Make sure that only pivot tables are selected. If any other objects are selected, the Range option is not available.
2. From the menus choose:
Edit > Find
3. Click the Range tab.
4. Select the type of range: Between, Greater than or equal to, or Less than or equal to.
5. Select the value or values that define the range.
 - If either value contains non-numeric characters, both values are treated as strings.
 - If both values are numbers, only numeric values are searched.
 - You cannot use the Range tab to replace values.

This feature is not available for legacy tables. See the topic [Legacy tables](#) for more information.

Copying output into other applications

Output objects can be copied and pasted into other applications, such as a word-processing program or a spreadsheet. You can paste output in a variety of formats. Depending on the target application and the selected output objects, some or all of the following formats may be available:

Metafile. WMF and EMF metafile format. These formats are available only on Windows operating systems.

RTF (rich text format). Multiple selected objects, text output, and pivot tables can be copied and pasted in RTF format. For pivot tables, in most applications this means that the tables are pasted as tables that can then be edited in the other application. Pivot tables that are too wide for the document width will either be wrapped, scaled down to fit the document width, or left unchanged, depending on the pivot table options settings. See the topic [Pivot table options](#) for more information.

Note: Microsoft Word may not display extremely wide tables properly.

Image. JPG and PNG image formats.

BIFF. Pivot tables and text output can be pasted into a spreadsheet in BIFF format. Numbers in pivot tables retain numeric precision. This format is available only on Windows operating systems.

Text. Pivot tables and text output can be copied and pasted as text. This process can be useful for applications such as e-mail, where the application can accept or transmit only text.

Microsoft Office Graphic Object. Charts that support this format can be copied to Microsoft Office applications and edited in those applications as native Microsoft Office charts. Because of differences between SPSS® Statistics/SPSS Modeler charts and Microsoft Office charts, some features of SPSS Statistics/SPSS Modeler charts are not retained in the copied version. Copying multiple selected charts in Microsoft Office Graphic Object format is not supported.

If the target application supports multiple available formats, it may have a Paste Special menu item that allows you to select the format, or it may automatically display a list of available formats.

Note: Microsoft Office version 16 (or higher) is required when copying and pasting Boxplots and Histograms.

Copying and pasting multiple output objects

The following limitations apply when pasting multiple output objects into other applications:

- RTF format. In most applications, pivot tables are pasted as tables that can be edited in that application. Charts, trees, and model views are pasted as images.
- Metafile and image formats. All the selected output objects are pasted as a single object in the other application.
- BIFF format. Charts, trees, and model views are excluded.

Copy special

When copying and pasting large amounts of output, particularly very large pivot tables, you can improve the speed of the operation by using Edit > Copy Special to limit the number of formats copied to the clipboard.

You can also save the selected formats as the default set of formats to copy to the clipboard. This setting persists across sessions.

Copy as

You can right-click a selected object in the Output Viewer and select Edit > Copy as to copy to the most popular formats (for example, All, Image, or Microsoft Office Graphic Object). Selecting Edit > Copy copies All. Note that if Copy as is grayed out or not present for a selected object, this copy format is not available for that particular object.

Interactive output

Interactive output objects contain multiple, related output objects. The selection in one object can change what is displayed or highlighted in the other object. For example, selecting a row in a table might highlight an area in a map or display a chart for a different category.

Interactive output objects do not support editing features, such as changing text, colors, fonts, or table borders. The individual objects can be copied from the interactive object to the Viewer. Tables copied from interactive output can be edited in the pivot table editor.

Copying objects from interactive output

File>Copy to Viewer copies individual output objects to the Viewer window.

- The available options depend on the contents of the interactive output.
- Chart and Map create chart objects.
- Table creates a pivot table that can be edited in the pivot table editor.
- Snapshot creates an image of the current view.
- Model creates a copy of the current interactive output object.

Edit>Copy Object copies individual output objects to the clipboard.

- Pasting the copied object into the Viewer is equivalent to File>Copy to Viewer.
- Pasting the object into another application pastes the object as an image.

Zoom and Pan

For maps, you can use View>Zoom to zoom the view of the map. Within a zoomed map view, you can use View>Pan to move the view.

Print settings

File>Print Settings controls how interactive objects are printed.

- Print visible view only. Prints only the view that is currently displayed. This option is the default setting.
- Print all views. Prints all views contained in the interactive output.
- The selected option also determines the default action for exporting the output object.

Export output

Export Output saves Viewer output in HTML, text, Word/RTF, Excel, PowerPoint (requires PowerPoint 97 or later), and PDF formats. Charts can also be exported in a number of different graphics formats.

Note: Export to PowerPoint is available only on Windows operating systems.

Exporting Output

1. Make the Viewer the active dialog (click anywhere in the dialog).
2. Click the Export button on the toolbar or right-click in the output window and select Export.
3. Enter a file name (or prefix for charts) and select an export format.

Objects to Export. You can export all objects in the Viewer, all visible objects, or only selected objects.

Document Type. The available options are:

- Word/RTF (*.doc). Pivot tables are exported as Word tables with all formatting attributes intact (for example, cell borders, font styles, and background colors). Text output is exported as formatted RTF. Charts, tree diagrams, and model views are included in PNG format. Note that Microsoft Word might not display extremely wide tables properly.
- Excel 97-2004 (*.xls)/Excel 2007 and Higher (*.xlsx). Pivot table rows, columns, and cells are exported as Excel rows, columns, and cells, with all formatting attributes intact (for example, cell borders, font styles, and background colors). Text output is exported with all font attributes intact. Each line in the text output is a row in the Excel file, with the entire contents of the line in a single cell. Charts, tree diagrams, and model views are included in PNG format. Output can be exported as *Excel 97-2004* or *Excel 2007 and higher*.
- HTML (*.htm). Pivot tables are exported as HTML tables. Text output is exported as preformatted HTML. Charts, tree diagrams, and model views are embedded in the document in the selected graphic format. A browser compatible with HTML 5 is required for viewing output that is exported in HTML format.
- Portable Document Format (*.pdf). All output is exported as it appears in Print Preview, with all formatting attributes intact.
- Text - Plain/UTF8/UTF16 (*.txt). Text output formats include plain text, UTF-8, and UTF-16. Pivot tables can be exported in tab-separated or space-separated format. All text output is exported in space-separated format. For charts, tree diagrams, and model views, a line is inserted in the text file for each graphic, indicating the image file name.
- None (Graphics Only). Available export formats include: EPS, JPEG, TIFF, PNG, and BMP. On Windows operating systems, EMF (enhanced metafile) format is also available.

Open the containing folder. Opens the folder that contains the files that are created by the export.

- [HTML options](#)
 - [Web report options](#)
 - [Word options](#)
 - [Excel options](#)
 - [PowerPoint options](#)
 - [PDF options](#)
 - [Text options](#)
 - [Images only options](#)
 - [Graphics format options](#)
-

HTML options

HTML export requires a browser that is compatible with HTML 5.

The following options are available for exporting output in HTML format:

Layers in pivot tables. By default, inclusion or exclusion of pivot table layers is controlled by the table properties for each pivot table. You can override this setting and include all layers or exclude all but the currently visible layer. See the topic [Table properties: printing](#) for more information.

Export layered tables as interactive. Layered tables are displayed as they appear in the Viewer, and you can interactively change the displayed layer in the browser. If this option is not selected, each table layer is displayed as a separate table.

Tables as HTML. This controls style information included for exported pivot tables.

- **Export with styles and fixed column width.** All pivot table style information (font styles, background colors, etc.) and column widths are preserved.
- **Export without styles.** Pivot tables are converted to default HTML tables. No style attributes are preserved. Column width is determined automatically.

Include footnotes and captions. Controls the inclusion or exclusion of all pivot table footnotes and captions.

Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. See the topic [Model properties](#) for more information. (Note: all model views, including tables, are exported as graphics.)

Note: For HTML, you can also control the image file format for exported charts. See the topic [Graphics format options](#) for more information.

To set HTML export options

1. Make the Viewer the active window (click anywhere in the window).
 2. From the menus choose:
File > Export...
 3. Select HTML as the export format.
 4. Click Change Options.
-

Web report options

A web report is an interactive document that is compatible with most browsers. Many of the interactive features of pivot tables available in the Viewer are also available in web reports.

Report Title. The title that is displayed in the header of the report. By default, the file name is used. You can specify a custom title to use instead of the file name.

Format. There are two options for report format:

- SPSS Web Report (HTML 5). This format requires a browser that is compatible with HTML 5.
- Cognos Active Report (mht). This format requires a browser that supports MHT format files or the Cognos Active Report application.

Exclude Objects. You can exclude selected object types from the report:

- Text. Text objects that are not logs. This option includes text objects that contain information about the active dataset.
- Logs. Text objects that contain a listing of the command syntax that was run. Log items also include warnings and error messages that are encountered by commands that do not produce any Viewer output.
- Notes Tables. Output from statistical and charting procedures includes a Notes table. This table contains information about the dataset that was used, missing values, and the command syntax that was used to run the procedure.
- Warnings and Error Messages. Warnings and error messages from statistical and charting procedures.

Restyle the tables and charts to match the Web Report. This option applies the standard web report style to all tables and charts. This setting overrides any fonts, colors, or other styles in the output as displayed in the Viewer. You cannot modify the standard web report style.

Web Server Connection. You can include the URL location of one or more application servers that are running the IBM® SPSS® Statistics Web Report Application Server. The web application server provides features to pivot tables, edit charts, and save modified web reports.

- Select Use for each application server that you want to include in the web report.
- If a web report contains a URL specification, the web report connects to that application server to provide the additional editing features.
- If you specify multiple URLs, the web report attempts to connect to each server in the order in which they are specified.

The IBM SPSS Statistics Web Report Application Server can be downloaded from <http://www.ibm.com/developerworks/spssdevcentral>.

Related information

- [Export output](#)
 - [HTML Options](#)
-

Word options

The following options are available for exporting output in Word format:

Preserve break points. If you have defined break points, these settings will be preserved in the Word tables.

Include footnotes and captions. Controls the inclusion or exclusion of all pivot table footnotes and captions.

Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. See the topic [Model properties](#) for more information. (Note: all model views, including tables, are exported as graphics.)

Page Setup for Export. This opens a dialog where you can define the page size and margins for the exported document. The document width used to determine wrapping and shrinking behavior is the page width minus the left and right margins.

To set Word export options

1. Make the Viewer the active window (click anywhere in the window).
 2. From the menus choose:
File > Export...
 3. Select Word/RTF as the export format.
 4. Click Change Options.
-

Excel options

The following options are available for exporting output in Excel format:

Create a worksheet or workbook or modify an existing worksheet. By default, a new workbook is created. If a file with the specified name already exists, it will be overwritten. If you select the option to create a worksheet, if a worksheet with the specified name already exists in the specified file, it will be overwritten. If you select the option to modify an existing worksheet, you must also specify the worksheet name. (This is optional for creating a worksheet.) Worksheet names cannot exceed 31 characters and cannot contain forward or back slashes, square brackets, question marks, or asterisks.

When exporting to Excel 97-2004, if you modify an existing worksheet, charts, model views, and tree diagrams are not included in the exported output.

Location in worksheet. Controls the location within the worksheet for the exported output. By default, exported output will be added after the last column that has any content, starting in the first row, without modifying any existing contents. This is a good choice for adding new columns to an existing worksheet. Adding exported output after the last row is a good choice for adding new rows to an existing worksheet. Adding exported output starting at a specific cell location will overwrite any existing content in the area where the exported output is added.

Layers in pivot tables. By default, inclusion or exclusion of pivot table layers is controlled by the table properties for each pivot table. You can override this setting and include all layers or exclude all but the currently visible layer. See the topic [Table properties: printing](#) for more information.

Include footnotes and captions. Controls the inclusion or exclusion of all pivot table footnotes and captions.

Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. See the topic [Model properties](#) for more information. (Note: all model views, including tables, are exported as graphics.)

To set Excel export options

1. Make the Viewer the active window (click anywhere in the window).
 2. From the menus choose:
File > Export...
 3. Select Excel as the export format.
 4. Click Change Options.
-

PowerPoint options

The following options are available for PowerPoint:

Layers in pivot tables. By default, inclusion or exclusion of pivot table layers is controlled by the table properties for each pivot table. You can override this setting and include all layers or exclude all but the currently visible layer. See the topic [Table properties: printing](#) for more information.

Wide Pivot Tables. Controls the treatment of tables that are too wide for the defined document width. By default, the table is wrapped to fit. The table is divided into sections, and row labels are repeated for each section of the table. Alternatively, you can shrink wide tables or make no changes to wide tables and allow them to extend beyond the defined document width.

Include footnotes and captions. Controls the inclusion or exclusion of all pivot table footnotes and captions.

Use Viewer outline entries as slide titles. Includes a title on each slide that is created by the export. Each slide contains a single item that is exported from the Viewer. The title is formed from the outline entry for the item in the outline pane of the Viewer.

Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. See the topic [Model properties](#) for more information. (Note: all model views, including tables, are exported as graphics.)

Page Setup for Export. This opens a dialog where you can define the page size and margins for the exported document. The document width used to determine wrapping and shrinking behavior is the page width minus the left and right margins.

To set PowerPoint export options

1. Make the Viewer the active window (click anywhere in the window).
2. From the menus choose:
File > *Export...*
3. Select PowerPoint as the export format.
4. Click Change Options.

Note: Export to PowerPoint is available only on Windows operating systems.

Related information

- [Export output](#)
 - [HTML options](#)
 - [Word options](#)
 - [Excel options](#)
 - [PDF options](#)
 - [Text options](#)
 - [Images only options](#)
 - [JPEG chart export options](#)
 - [BMP chart export options](#)
 - [PNG chart export options](#)
 - [EMF and TIFF chart export options](#)
 - [EPS chart export options](#)
-

PDF options

The following options are available for PDF:

Embed bookmarks. This option includes bookmarks in the PDF document that correspond to the Viewer outline entries. Like the Viewer outline pane, bookmarks can make it much easier to navigate documents with a large number of output objects.

Embed fonts. Embedding fonts ensures that the PDF document will look the same on all computers. Otherwise, if some fonts used in the document are not available on the computer being used to view (or print) the PDF document, font substitution may yield suboptimal results.

Layers in pivot tables. By default, inclusion or exclusion of pivot table layers is controlled by the table properties for each pivot table. You can override this setting and include all layers or exclude all but the currently visible layer. See the topic [Table properties: printing](#) for more information.

Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. See the topic [Model properties](#) for more information. (Note: all model views, including tables, are exported as graphics.)

To set PDF export options

1. Make the Viewer the active window (click anywhere in the window).
2. From the menus choose:
File > *Export...*
3. Select Portable Document Format as the export format.
4. Click Change Options.

Other Settings That Affect PDF Output

Page Setup/Page Attributes. Page size, orientation, margins, content and display of page headers and footers, and printed chart size in PDF documents are controlled by page setup and page attribute options.

Table Properties/TableLooks. Scaling of wide and/or long tables and printing of table layers are controlled by table properties for each table. These properties can also be saved in TableLooks.

Default/Current Printer. The resolution (DPI) of the PDF document is the current resolution setting for the default or currently selected printer (which can be changed using Page Setup). The maximum resolution is 1200 DPI. If the printer setting is higher, the PDF document resolution will be 1200 DPI.

Note: High-resolution documents may yield poor results when printed on lower-resolution printers.

Text options

Setting text output options

The following options are available for text export:

Pivot Table Format. Pivot tables can be exported in tab-separated or space-separated format. For space-separated format, you can also control:

- **Column Width.** Autofit does not wrap any column contents, and each column is as wide as the widest label or value in that column.
Custom sets a maximum column width that is applied to all columns in the table, and values that exceed that width wrap onto the next line in that column.
- **Row/Column Border Character.** Controls the characters used to create row and column borders. To suppress display of row and column borders, enter blank spaces for the values.

Layers in pivot tables. By default, inclusion or exclusion of pivot table layers is controlled by the table properties for each pivot table. You can override this setting and include all layers or exclude all but the currently visible layer. See the topic [Table properties: printing](#) for more information.

Include footnotes and captions. Controls the inclusion or exclusion of all pivot table footnotes and captions.

Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. See the topic [Model properties](#) for more information. (Note: all model views, including tables, are exported as graphics.)

To set text export options

1. Make the Viewer the active window (click anywhere in the window).
2. From the menus choose:
File > Export...
3. Select Text as the export format.
4. Click Change Options.

Images only options

The following options are available for exporting images only:

1. Make the Workbook tab the active tab (click anywhere in the tab).
2. Select the items that you want to export, click the vertical ellipsis control, and then select Export from the menu. You can also click the Export button on the toolbar. The Export workbook dialog displays.

Document setting

Type

Select Images only as the document Type.

Image format

Select an image format for exported images.

PNG

When selected, images are exported in Portable Network Graphics (*.png) format. The setting provides options for selecting the image Scale (%) and Color depth.

JPG

When selected, images are exported in Joint Photographic Experts Group (*.jpg) format. The setting provides options for selecting the image Scale (%) and Convert to grayscale.

TIFF

When selected, images are exported in Tag Image File Format (*.tiff) format. The setting provides options for selecting the image Scale (%).

BMP

When selected, images are exported in bitmap (*.bmp) format. The setting provides options for selecting the image Scale (%) and Compress image to reduce file size.

Open the containing folder

Optionally, select the setting to open the export save file location folder after export.

Objects to export

Objects to include

Select to export All or Selected (the default setting) output objects.

3. Click Export after specifying the image only export settings.
-

Graphics format options

For HTML and text documents and for exporting charts only, you can select the graphic format, and for each graphic format you can control various optional settings.

To select the graphic format and options for exported charts:

1. Make the Viewer the active window (click anywhere in the window).
2. From the menus choose:
File > Export...
3. Select HTML, Text, or None (Graphics only) as the document type.
4. Select the graphic file format from the drop-down list.
5. Click Change Options to change the options for the selected graphic file format.
 - [JPEG chart export options](#)
 - [BMP chart export options](#)
 - [PNG chart export options](#)
 - [EMF and TIFF chart export options](#)
 - [EPS chart export options](#)

JPEG chart export options

Image size(%)

Percentage of original chart size, up to 200 percent.

Convert to grayscale

Converts colors to shades of gray.

BMP chart export options

Image size(%)

Percentage of original chart size, up to 200 percent.

Compress image to reduce file size

A lossless compression technique that creates smaller files without affecting image quality.

PNG chart export options

Image size(%)

Percentage of original chart size, up to 200 percent.

Color Depth

Determines the number of colors in the exported chart. A chart that is saved under any depth will have a minimum of the number of colors that are actually used and a maximum of the number of colors that are allowed by the depth. For example, if the chart contains three colors (red, white, and black) and you save it as 16 colors, the chart will remain as three colors.

Note: If the number of colors in the chart exceeds the number of colors for that depth, the colors will be dithered to replicate the colors in the chart.

EMF and TIFF chart export options

Image size. Percentage of original chart size, up to 200 percent.

Note: EMF (enhanced metafile) format is available only on Windows operating systems.

Related information

- [Graphics format options](#)
 - [Export output](#)
 - [HTML options](#)
 - [Word options](#)
 - [Excel options](#)
 - [PowerPoint options](#)
 - [PDF options](#)
 - [Text options](#)
 - [Images only options](#)
 - [JPEG chart export options](#)
 - [BMP chart export options](#)
 - [PNG chart export options](#)
 - [EPS chart export options](#)
-

EPS chart export options

Image size. You can specify the size as a percentage of the original image size (up to 200 percent), or you can specify an image width in pixels (with height determined by the width value and the aspect ratio). The exported image is always proportional to the original.

Include TIFF preview image. Saves a preview with the EPS image in TIFF format for display in applications that cannot display EPS images on screen.

Fonts. Controls the treatment of fonts in EPS images.

- **Use font references.** If the fonts that are used in the chart are available on the output device, the fonts are used. Otherwise, the output device uses alternate fonts.
- **Replace fonts with curves.** Turns fonts into PostScript curve data. The text itself is no longer editable as text in applications that can edit EPS graphics. This option is useful if the fonts that are used in the chart are not available on the output device.

Related information

- [Graphics format options](#)
 - [Export output](#)
 - [HTML options](#)
 - [Word options](#)
 - [Excel options](#)
 - [PowerPoint options](#)
 - [PDF options](#)
 - [Text options](#)
 - [Images only options](#)
 - [JPEG chart export options](#)
 - [BMP chart export options](#)
 - [PNG chart export options](#)
 - [EMF and TIFF chart export options](#)
-

Viewer printing

There are two options for printing the contents of the Viewer window:

All visible output. Prints only items that are currently displayed in the contents pane. Hidden items (items with a closed book icon in the outline pane or hidden in collapsed outline layers) are not printed.

Selection. Prints only items that are currently selected in the outline and/or contents panes.

- [To print output and charts](#)
- [Print Preview](#)
- [Page Attributes: Headers and Footers](#)
- [Page Attributes: Options](#)

Related information

- [To print output and charts](#)
 - [Page Attributes: Options](#)
 - [Print Preview](#)
-

To print output and charts

1. Make the Viewer the active window (click anywhere in the window).
2. From the menus choose:
File > Print...
3. Select the print settings that you want.
4. Click OK to print.

Related information

- [Viewer printing](#)
 - [Page Attributes: Options](#)
-

Print Preview

Print Preview shows you what will print on each page for Viewer documents. It is a good idea to check Print Preview before actually printing a Viewer document, because Print Preview shows you items that may not be visible by looking at the contents pane of the Viewer, including:

- Page breaks
- Hidden layers of pivot tables
- Breaks in wide tables
- Headers and footers that are printed on each page

If any output is currently selected in the Viewer, the preview displays only the selected output. To view a preview for all output, make sure nothing is selected in the Viewer.

Related information

- [Viewer printing](#)
-

Page Attributes: Headers and Footers

Headers and footers are the information that is printed at the top and bottom of each page. You can enter any text that you want to use as headers and footers. You can also use the toolbar in the middle of the dialog box to insert:

- Date and time
 - Page numbers
 - Viewer filename
 - Outline heading labels
 - Page titles and subtitles
 - Make Default uses the settings specified here as the default settings for new Viewer documents. (Note: this makes the current settings on both the Header/Footer tab and the Options tab the default settings.)
 - Outline heading labels indicate the first-, second-, third-, and/or fourth-level outline heading for the first item on each page.
 - Page titles and subtitles print the current page titles and subtitles. These can be created with New Page Title on the Viewer Insert menu or with the **TITLE** and **SUBTITLE** commands. If you have not specified any page titles or subtitles, this setting is ignored.
- Note:* Font characteristics for new page titles and subtitles are controlled on the Viewer tab of the Options dialog box (accessed by choosing Options on the Edit menu). Font characteristics for existing page titles and subtitles can be changed by editing the titles in the Viewer.

To see how your headers and footers will look on the printed page, choose Print Preview from the File menu.

- [To insert page headers and footers](#)

Related information

- [To insert page headers and footers](#)
 - [Page Attributes: Options](#)
-

To insert page headers and footers

1. Make the Viewer the active window (click anywhere in the window).
 2. From the menus, choose:
File > Header and Footer...
 3. Enter the header and/or footer that you want to appear on each page.
-

Page Attributes: Options

This dialog box controls the printed chart size, the space between printed output items, and page numbering.

- Printed Chart Size. Controls the size of the printed chart relative to the defined page size. The chart's aspect ratio (width-to-height ratio) is not affected by the printed chart size. The overall printed size of a chart is limited by both its height and width. When the outer borders of a chart reach the left and right borders of the page, the chart size cannot increase further to fill more page height.
- Space between items. Controls the space between printed items. Each pivot table, chart, and text object is a separate item. This setting does not affect the display of items in the Viewer.
- Number pages starting with. Numbers pages sequentially, starting with the specified number.
- Make Default. This option uses the settings that are specified here as the default settings for new Viewer documents. (Note that this option makes the current Header/Footer settings and Options settings the default.)
- [To change printed chart size, page numbering, and space between printed Items](#)

Related information

- [To change printed chart size, page numbering, and space between printed Items](#)
 - [Viewer printing](#)
 - [To print output and charts](#)
 - [Page Attributes: Headers and Footers](#)
-

To change printed chart size, page numbering, and space between printed Items

1. Make the Viewer the active window (click anywhere in the window).
 2. From the menus choose:
File > Page Attributes...
 3. Click the Options tab.
 4. Change the settings and click OK.
-

Saving output

The contents of the Viewer can be saved.

- Output object (*.cou). This format saves the entire output container, including the graph, tabs, annotations, and so on. This format can be opened and viewed in IBM® SPSS® Modeler, added to projects, and published and tracked using the IBM SPSS Collaboration and Deployment Services Repository. This format is not compatible with IBM SPSS Statistics.
- Viewer files (*.spv). The format that is used to display files in the Viewer window. When you save to this format from a model nugget in IBM SPSS Modeler, only the content of the Viewer from the Model tab is saved.

To control options for saving web reports or save results in other formats (for example, text, Word, Excel), use Export on the File menu.

- [Saving a Viewer document](#)

Related information

- [Export output](#)
 - [Publish to Web](#)
 - [Viewer](#)
-

Saving a Viewer document

1. From the Viewer window menus choose:

File > Save

2. Enter the name of the document, and then click Save.

Optionally, you can do the following:

Lock files to prevent editing in IBM® SPSS® Smartreader

If a Viewer document is locked, you can manipulate pivot tables (swap rows and columns, change the displayed layer, etc.) but you cannot edit any output or save any changes to the Viewer document in IBM SPSS Smartreader (a separate product for working with Viewer documents). This setting has no effect on Viewer documents opened in IBM SPSS Statistics or IBM SPSS Modeler.

Encrypt files with a password

You can protect confidential information stored in a Viewer document by encrypting the document with a password. Once encrypted, the document can only be opened by providing the password. IBM SPSS Smartreader users will also be required to provide the password in order to open the file.

To encrypt a Viewer document:

- a. Select Encrypt file with password in the Save Output As dialog box.
- b. Click Save.
- c. In the Encrypt File dialog box, provide a password and re-enter it in the Confirm password text box. Passwords are limited to 10 characters and are case-sensitive.

Note: Passwords cannot be recovered if they are lost. If the password is lost the file cannot be opened.

Creating strong passwords

- Use eight or more characters.
- Include numbers, symbols and even punctuation in your password.
- Avoid sequences of numbers or characters, such as "123" and "abc", and avoid repetition, such as "111aaa".
- Do not create passwords that use personal information such as birthdays or nicknames.
- Periodically change the password.

Note: Storing encrypted files to an IBM SPSS Collaboration and Deployment Services Repository is not supported.

Modifying encrypted files

- If you open an encrypted file, make modifications to it and choose File > Save, the modified file will be saved with the same password.
- You can change the password on an encrypted file by opening the file, repeating the steps for encrypting it, and specifying a different password in the Encrypt File dialog box.
- You can save an unencrypted version of an encrypted file by opening the file, choosing File > Save As and deselecting Encrypt file with password in the Save Output As dialog box.

Note: Encrypted data files and output documents cannot be opened in versions of IBM SPSS Statistics prior to version 21.

Encrypted syntax files cannot be opened in versions prior to version 22.

Store required model information with the output document

This option applies only when there are model viewer items in the output document that require auxiliary information to enable some of the interactive features. Click More Info to display a list of these model viewer items and the interactive features that require auxiliary information. Storing this information with the output document might substantially increase the document size. If you choose not to store this information, you can still open these output items but the specified interactive features will not be available.

Pivot tables

- [Pivot tables](#)
- [Manipulating a pivot table](#)
- [Working with layers](#)
- [Showing and hiding items](#)
- [TableLooks](#)

- [Table properties](#)
 - [Cell properties](#)
 - [Footnotes and captions](#)
 - [Data cell widths](#)
 - [Changing column width](#)
 - [Displaying hidden borders in a pivot table](#)
 - [Selecting rows, columns, and cells in a pivot table](#)
 - [Printing pivot tables](#)
 - [Creating a chart from a pivot table](#)
 - [Legacy tables](#)
-

Pivot tables

Many results are presented in tables that can be pivoted interactively. That is, you can rearrange the rows, columns, and layers.

Manipulating a pivot table

Options for manipulating a pivot table include:

- Transposing rows and columns
 - Moving rows and columns
 - Creating multidimensional layers
 - Grouping and ungrouping rows and columns
 - Showing and hiding rows, columns, and other information
 - Rotating row and column labels
 - Finding definitions of terms
 - [Activating a pivot table](#)
 - [Pivoting a table](#)
 - [Changing display order of elements within a dimension](#)
 - [Moving rows and columns within a dimension element](#)
 - [Transposing rows and columns](#)
 - [Grouping rows or columns](#)
 - [Ungrouping rows or columns](#)
 - [Rotating row or column labels](#)
 - [Sorting rows](#)
 - [Inserting rows and columns](#)
 - [Controlling display of variable and value labels](#)
 - [Changing the output language](#)
 - [Navigating large tables](#)
 - [Undoing changes](#)
-

Activating a pivot table

Before you can manipulate or modify a pivot table, you need to **activate** the table. To activate a table:

1. Double-click the table.
or
 2. Right-click the table and from the pop-up menu choose Edit Content.
 3. From the sub-menu choose either In Viewer or In Separate Window.
-

Pivoting a table

A table has three dimensions: rows, columns, and layers. A dimension can contain multiple elements (or none at all). You can change the organization of the table by moving elements between or within dimensions. To move an element, just drag and drop it where you want it.

Related information

- [Activating a pivot table](#)
 - [Changing display order of elements within a dimension](#)
 - [Moving rows and columns within a dimension element](#)
 - [Transposing rows and columns](#)
-

Changing display order of elements within a dimension

To change the display order of elements within a table dimension (row, column, or layer):

1. Activate the pivot table.
 2. If pivoting trays are not already on, from the Pivot Table menu choose:
Pivot > Pivoting Trays
 3. Drag and drop the elements within the dimension in the pivoting tray.
-

Moving rows and columns within a dimension element

1. Activate the pivot table.
2. In the table itself (not the pivoting trays), click the label for the row or column you want to move.
3. Drag the label to the new position.

Related information

- [Activating a pivot table](#)
-

Transposing rows and columns

1. Activate the pivot table.
2. From the menus choose:
Pivot > Transpose Rows and Columns

This has the same effect as dragging all of the row elements into the column dimension and dragging all of the column elements into the row dimension.

Grouping rows or columns

1. Activate the pivot table.
2. Select the labels for the rows or columns that you want to group together (click and drag or Shift+click to select multiple labels).
3. From the menus choose:
Edit > Group

A group label is automatically inserted. Double-click the group label to edit the label text.

Note: To add rows or columns to an existing group, you must first ungroup the items that are currently in the group. Then you can create a new group that includes the additional items.

Related information

- [Manipulating a pivot table](#)
- [Activating a pivot table](#)
- [Pivoting a table](#)
- [Changing display order of elements within a dimension](#)
- [Moving rows and columns within a dimension element](#)
- [Transposing rows and columns](#)
- [Ungrouping rows or columns](#)
- [Rotating row or column labels](#)
- [Undoing changes](#)

Ungrouping rows or columns

Ungrouping automatically deletes the group label.

Related information

- [Manipulating a pivot table](#)
- [Activating a pivot table](#)
- [Pivoting a table](#)
- [Changing display order of elements within a dimension](#)
- [Moving rows and columns within a dimension element](#)
- [Transposing rows and columns](#)
- [Grouping rows or columns](#)
- [Rotating row or column labels](#)
- [Undoing changes](#)

Rotating row or column labels

You can rotate labels between horizontal and vertical display for the innermost column labels and the outermost row labels in a table.

1. Activate the pivot table.
2. From the menus choose:
Format > Rotate Inner Column Labels
or
Format > Rotate Outer Row Labels

Only the innermost column labels and the outermost row labels can be rotated.

Related information

- [Manipulating a pivot table](#)
- [Activating a pivot table](#)
- [Pivoting a table](#)
- [Changing display order of elements within a dimension](#)
- [Moving rows and columns within a dimension element](#)
- [Transposing rows and columns](#)
- [Grouping rows or columns](#)
- [Ungrouping rows or columns](#)
- [Undoing changes](#)

Sorting rows

To sort the rows of a pivot table:

1. Activate the table.
2. Select any cell in the column you want to use to sort on. To sort just a selected group of rows, select two or more contiguous cells in the column you want to use to sort on.
3. From the menus, choose:
Edit > Sort rows
4. Select Ascending or Descending from the submenu.
 - If the row dimension contains groups, sorting affects only the group that contains the selection.
 - You cannot sort across group boundaries.
 - You cannot sort tables with more than one item in the row dimension.

Inserting rows and columns

To insert a row or column in a pivot table:

1. Activate the table.
2. Select any cell in the table.
3. From the menus, choose:

Insert Before

or

Insert After

From the submenu, choose:

Row

or

Column

- A plus sign (+) is inserted in each cell of the new row or column to prevent the new row or column from being automatically hidden because it is empty.
 - In a table with nested or layered dimensions, a column or row is inserted at every corresponding dimension level.
-

Controlling display of variable and value labels

If variables contain descriptive variable or value labels, you can control the display of variable names and labels and data values and value labels in pivot tables.

1. Activate the pivot table.
2. From the menus, choose:

View > Variable labels

or

View > Value labels

3. Select one of the follow options from the submenu:

- Name or Value. Only variable names (or values) are displayed. Descriptive labels are not displayed.
 - Label. Only descriptive labels are displayed. Variable names (or values) are not displayed.
 - Both. Both names (or values) and descriptive labels are displayed.
-

Changing the output language

To change the output language in a pivot table:

1. Activate the table
2. From the menus, choose:

View > Language

3. Select one of the available languages.

Changing the language affects only text that is generated by the application, such as table titles, row and column labels, and footnote text. Variable names and descriptive variable and value labels are not affected.

Navigating large tables

To use the navigation window to navigate large tables:

1. Activate the table.
2. From the menus choose:

View > Navigation

Undoing changes

You can undo the most recent change or all changes to an activated pivot table. Both actions apply only to changes made since the most recent activation of the table.

To undo the most recent change:

1. From the menus, choose:
Edit > Undo

To undo all changes:

2. From the menus, choose:
Edit > Restore

Working with layers

You can display a separate two-dimensional table for each category or combination of categories. The table can be thought of as stacked in layers, with only the top layer visible.

- [Creating and displaying layers](#)
- [Go to layer category](#)

Creating and displaying layers

To create layers:

1. If pivoting trays are not already on, from the Pivot Table menu, choose:
Pivot > Pivoting Trays
2. Drag an element from the row or column dimension into the layer dimension.

Moving elements to the layer dimension creates a multidimensional table, but only a single two-dimensional "slice" is displayed. The visible table is the table for the top layer. For example, if a yes/no categorical variable is in the layer dimension, then the multidimensional table has two layers: one for the "yes" category and one for the "no" category.

Changing the displayed layer

1. Choose a category from the drop-down list of layers (in the pivot table itself, not the pivoting tray).
or
2. From the menus, choose:
Pivot > Go to Layer...

Go to layer category

Go to Layer Category allows you to change layers in a pivot table. This dialog box is particularly useful when there are many layers or the selected layer has many categories.

Related information

- [Working with layers](#)
- [Creating and displaying layers](#)

Showing and hiding items

Many types of cells can be hidden, including:

- Dimension labels
 - Categories, including the label cell and data cells in a row or column
 - Category labels (without hiding the data cells)
 - Footnotes, titles, and captions
- [Hiding rows and columns in a table](#)
 - [Showing hidden rows and columns in a table](#)
 - [Hiding and showing dimension labels](#)
 - [Hiding and showing table titles](#)
-

Hiding rows and columns in a table

Showing hidden rows and columns in a table

1. Activate the pivot table.
2. From the menus choose:
View > Show All Categories

This displays all hidden rows and columns in the table. (If Hide empty rows and columns is selected in Table Properties for this table, a completely empty row or column remains hidden.)

Related information

- [Showing and hiding items](#)
 - [Hiding rows and columns in a table](#)
 - [Hiding and showing dimension labels](#)
 - [Hiding and showing table titles](#)
-

Hiding and showing dimension labels

1. Activate the pivot table.
2. Select the dimension label or any category label within the dimension.
3. From the View menu or the pop-up menu choose Hide Dimension Label or Show Dimension Label.

Related information

- [Showing and hiding items](#)
 - [Hiding rows and columns in a table](#)
 - [Showing hidden rows and columns in a table](#)
 - [Hiding and showing table titles](#)
-

Hiding and showing table titles

To hide a title:

1. Activate the pivot table.
2. Select the title.
3. From the View menu choose Hide.

To show hidden titles:

4. From the View menu choose Show All.

Related information

- [Showing and hiding items](#)
- [Hiding rows and columns in a table](#)

- [Showing hidden rows and columns in a table](#)
 - [Hiding and showing dimension labels](#)
-

TableLooks

A TableLook is a set of properties that define the appearance of a table. You can select a previously defined TableLook or create your own TableLook.

- Before or after a TableLook is applied, you can change cell formats for individual cells or groups of cells by using cell properties. The edited cell formats remain intact, even when you apply a new TableLook.
 - Optionally, you can reset all cells to the cell formats that are defined by the current TableLook. This option resets any cells that were edited. If As Displayed is selected in the TableLook Files list, any edited cells are reset to the current table properties.
 - Only table properties that are defined in the Table Properties dialog are saved in TableLooks. TableLooks do not include individual cell modifications.
- [To apply a TableLook](#)
[To edit or create a TableLook](#)
-

To apply a TableLook

1. Activate a pivot table.
2. From the menus, choose:
Format > TableLooks...
3. Select a TableLook from the list of files. To select a file from another directory, click Browse.
4. Click OK to apply the TableLook to the selected pivot table.

You can also edit TableLooks and create new ones.

To edit or create a TableLook

1. Activate a pivot table.
 2. From the menus, choose:
Format > TableLooks...
 3. In the TableLooks dialog box, select a TableLook from the list of files.
 4. Click Edit Look.
 5. Adjust the table properties for the attributes that you want, and then click OK.
 6. Click Save Look to save the edited TableLook, or click Save As to save it as a new TableLook.
- Editing a TableLook affects only the selected pivot table. An edited TableLook is not applied to any other tables that use that TableLook unless you select those tables and reapply the TableLook. For information on setting the default TableLook applied to all new tables, see [Setting the TableLook for new pivot tables](#).
 - Only table properties that are defined in the Table Properties dialog are saved in TableLooks. TableLooks do not include individual cell modifications.
-

Table properties

Table Properties allows you to set general properties of a table, and set cell styles for various parts of a table. You can:

- Control general properties, such as hiding empty rows or columns and adjusting printing properties.
 - Control the format and position of footnote markers.
 - Determine specific formats for cells in the data area, for row and column labels, and for other areas of the table.
 - Control the width and color of the lines that form the borders of each area of the table.
- [To change pivot table properties](#)
[Table properties: general](#)
[Table properties: notes](#)
[Table properties: cell formats](#)
[Table properties: borders](#)

- [Table properties: printing](#)

Related information

- [Pivot tables](#)
- [Manipulating a pivot table](#)
- [Working with layers](#)
- [Showing and hiding items](#)
- [TableLooks](#)
- [Cell properties](#)
- [Footnotes and captions](#)
- [Data cell widths](#)
- [Changing column width](#)
- [Selecting rows, columns, and cells in a pivot table](#)
- [Printing pivot tables](#)
- [Set rows to display](#).

To change pivot table properties

1. From the menus choose:
Format > Table Properties...
2. Select a tab (General, Footnotes, Cell Formats, Borders, or Printing).
3. Select the options that you want.
4. Click OK or Apply.

The new properties are applied to the selected pivot table.

Table properties: general

Several properties apply to the table as a whole. You can:

- Show or hide empty rows and columns. (An empty row or column has nothing in any of the data cells.)
- Control the placement of row labels, which can be in the upper left corner or nested.
- Control maximum and minimum column width (expressed in points).

To change general table properties:

1. From the menus, choose:
Format > Table Properties...
2. Click the General tab.
3. Select the options that you want.
4. Click OK or Apply.

The new properties are applied to the selected pivot table.

- [Set rows to display](#)

Set rows to display

Note: This feature only applies to legacy tables.

By default, tables with many rows are displayed in sections of 100 rows. To control the number of rows displayed in a table:

1. Activate the table.
2. From the menus choose:
Table Properties > General
3. Select Display table by rows.
4. Click Set Rows to Display.
or

5. From the View menu of an activated pivot table, choose Display table by rows and Set Rows to Display.

Rows to display. Controls the maximum number of rows to display at one time. Navigation controls allow you move to different sections of the table. The minimum value is 10. The default is 100.

Widow/orphan tolerance. Controls the maximum number of rows of the inner most row dimension of the table to split across displayed views of the table. For example, if there are six categories in each group of the inner most row dimension, specifying a value of six would prevent any group from splitting across displayed views. This setting can cause the total number of rows in a displayed view to exceed the specified maximum number of rows to display.

Table properties: notes

The Notes tab of the Table Properties dialog controls footnote formatting and table comment text.

Footnotes. The properties of footnote markers include style and position in relation to text.

- The style of footnote markers is either numbers (1, 2, 3, ...) or letters (a, b, c, ...).
- The footnote markers can be attached to text as superscripts or subscripts.

Comment Text. You can add comment text to each table.

- Comment text is displayed in a tooltip when you hover over a table in the Viewer.
- Screen readers read the comment text when the table has focus.
- The tooltip in the Viewer displays only the first 200 characters of the comment, but screen readers read the entire text.
- When you export output to HTML, the comment text is used as alt text.

The new properties are applied to the selected pivot table. To apply new table properties to a TableLook instead of just the selected table, edit the TableLook (Format menu, TableLooks).

Table properties: cell formats

For formatting, a table is divided into areas: title, layers, corner labels, row labels, column labels, data, caption, and footnotes. For each area of a table, you can modify the associated cell formats. Cell formats include text characteristics (such as font, size, color, and style), horizontal and vertical alignment, background colors, and inner cell margins.

Cell formats are applied to areas (categories of information). They are not characteristics of individual cells. This distinction is an important consideration when pivoting a table.

For example,

- If you specify a bold font as a cell format of column labels, the column labels will appear bold no matter what information is currently displayed in the column dimension. If you move an item from the column dimension to another dimension, it does not retain the bold characteristic of the column labels.
- If you make column labels bold simply by highlighting the cells in an activated pivot table and clicking the Bold button on the toolbar, the contents of those cells will remain bold no matter what dimension you move them to, and the column labels will not retain the bold characteristic for other items moved into the column dimension.

To change cell formats:

1. From the menus choose:
Format > Table Properties...
2. Select the Cell Formats tab.
3. Select an Area from the drop-down list or click an area of the sample.
4. Select characteristics for the area. Your selections are reflected in the sample.
5. Click OK or Apply.

The new properties are applied to the selected pivot table.

Alternating row colors

To apply a different background and/or text color to alternate rows in the Data area of the table:

1. Select Data from the Area drop-down list.
2. Select (check) Alternate row color in the Background Color group.
3. Select the colors to use for the alternate row background and text.

Alternate row colors affect only the Data area of the table. They do not affect row or column label areas.

Table properties: borders

For each border location in a table, you can select a line style and a color. If you select None as the style, there will be no line at the selected location.

To change table borders:

1. From the menus choose:
Format > Table Properties...
2. Click the Borders tab.
3. Select a border location, either by clicking its name in the list or by clicking a line in the Sample area.
4. Select a line style or select None.
5. Select a color.
6. Click OK or Apply.

The new properties are applied to the selected pivot table.

Table properties: printing

You can control the following properties for printed pivot tables:

- Print all layers or only the top layer of the table, and print each layer on a separate page.
- Shrink a table horizontally or vertically to fit the page for printing.
- Control widow/orphan lines by controlling the minimum number of rows and columns that will be contained in any printed section of a table if the table is too wide and/or too long for the defined page size.
Note: If a table is too long to fit on the current page because there is other output above it, but it will fit within the defined page length, the table is automatically printed on a new page, regardless of the widow/orphan setting.
- Include continuation text for tables that don't fit on a single page. You can display continuation text at the bottom of each page and at the top of each page. If neither option is selected, the continuation text will not be displayed.

To control pivot table printing properties:

1. From the menus choose:
Format > Table Properties...
2. Click the Printing tab.
3. Select the printing options that you want.
4. Click OK or Apply.

The new properties are applied to the selected pivot table.

Related information

- [Table properties](#)
- [Table properties: general](#)
- [Set rows to display](#)
- [Table properties: notes](#)
- [Table properties: cell formats](#)
- [Table properties: borders](#)
- [Printing pivot tables](#)
- [Controlling table breaks for wide and long tables](#)

Cell properties

Cell properties are applied to a selected cell. You can change the font, value format, alignment, margins, and colors. Cell properties override table properties; therefore, if you change table properties, you do not change any individually applied cell properties.

To change cell properties:

1. Select the cells in the table.
2. From the Format menu or the pop-up menu choose Cell Properties.

- [Cell properties: Font and background](#)
 - [Cell properties: Format value](#)
 - [Cell properties: Alignment and margins](#)
-

Cell properties: Font and background

The Font and Background tab controls the font style and color and background color for the selected cells in the table.

To change font and background settings for selected cells in a pivot table:

Related information

- [Table properties: cell formats](#)
 - [Cell properties](#)
 - [Cell properties: Format value](#)
 - [Cell properties: Alignment and margins](#)
-

Cell properties: Format value

The Format Value tab controls value formats for the selected cells. You can select formats for numbers, dates, time, or currencies, and you can adjust the number of decimal digits that are displayed.

To change value formats for selected cells in a pivot table:

Related information

- [Table properties: cell formats](#)
 - [Cell properties](#)
 - [Cell properties: Font and background](#)
 - [Cell properties: Alignment and margins](#)
-

Cell properties: Alignment and margins

The Alignment and Margins tab controls horizontal and vertical alignment of values and top, bottom, left, and right margins for the selected cells. Mixed horizontal alignment aligns the content of each cell according to its type. For example, dates are right-aligned and text values are left-aligned.

To change alignment or margins for selected cells in a pivot table:

Related information

- [Table properties: cell formats](#)
 - [Cell properties](#)
 - [Cell properties: Font and background](#)
 - [Cell properties: Format value](#)
-

Footnotes and captions

You can add footnotes and captions to a table. You can also hide footnotes or captions, change footnote markers, and renumber footnotes.

- [Adding footnotes and captions](#)
- [To hide or show a caption](#)
- [To hide or show a footnote in a table](#)
- [Footnote marker](#)
- [Renumbering footnotes](#)
- [Editing footnotes in legacy tables](#)

Adding footnotes and captions

To add a caption to a table:

1. From the Insert menu choose Caption.

A footnote can be attached to any item in a table. To add a footnote:

1. Click a title, cell, or caption within an activated pivot table.
2. From the Insert menu choose Footnote.
3. Insert the footnote text in the provided area.

Related information

- [Table properties: notes](#)
 - [Footnotes and captions](#)
 - [To hide or show a caption](#)
 - [To hide or show a footnote in a table](#)
 - [Footnote marker](#)
 - [Renumbering footnotes](#)
 - [Editing footnotes in legacy tables](#)
 - [Footnote font and color settings](#)
-

To hide or show a caption

To hide a caption:

1. Activate the pivot table.
2. Select the caption.
3. From the View menu choose Hide.

To show hidden captions:

1. From the View menu choose Show All.

Related information

- [Table properties: notes](#)
 - [Footnotes and captions](#)
 - [Adding footnotes and captions](#)
 - [To hide or show a footnote in a table](#)
 - [Footnote marker](#)
 - [Renumbering footnotes](#)
 - [Editing footnotes in legacy tables](#)
 - [Footnote font and color settings](#)
-

To hide or show a footnote in a table

To hide a footnote:

1. Activate the pivot table.
2. Right-click the cell that contains the footnote reference and select Hide Footnotes from the pop-up menu
or

3. Select the footnote in the footnote area of the table and select Hide from the pop-up menu.

Note: For legacy tables, select the footnote area of the table, select Edit Footnote from the pop-up menu, and then deselect (clear) the Visible property for any footnotes you want to hide.

If a cell contains multiple footnotes, use the latter method to selectively hide footnotes.

To hide all footnotes in the table:

1. Select all of the footnotes in the footnote area of the table (use click and drag or Shift+click to select the footnotes) and select Hide from the View menu.

To show hidden footnotes:

1. Select Show All Footnotes from the View menu.

Related information

- [Table properties: notes](#)
 - [Footnotes and captions](#)
 - [Adding footnotes and captions](#)
 - [To hide or show a caption](#)
 - [Footnote marker](#)
 - [Renumbering footnotes](#)
 - [Editing footnotes in legacy tables](#)
 - [Footnote font and color settings](#)
-

Footnote marker

Footnote Marker changes the characters that can be used to mark a footnote. By default, standard footnote markers are sequential letters or numbers, depending on the table properties settings. You can also assign a special marker. Special markers are not affected when you renumber footnotes or switch between numbers and letters for standard markers. The display of numbers or letters for standard markers and the subscript or superscript position of footnote markers are controlled by the Footnotes tab of the Table Properties dialog.

To change footnote markers:

1. Select a footnote.
2. From the Format menu, choose Footnote Marker.

Special markers are limited to 2 characters. Footnotes with special markers precede footnotes with sequential letters or numbers in the footnote area of the table; so changing to a special marker can reorder the footnote list.

Related information

- [Table properties: notes](#)
 - [Footnotes and captions](#)
 - [Adding footnotes and captions](#)
 - [To hide or show a caption](#)
 - [To hide or show a footnote in a table](#)
 - [Renumbering footnotes](#)
 - [Editing footnotes in legacy tables](#)
 - [Footnote font and color settings](#)
-

Renumbering footnotes

When you have pivoted a table by switching rows, columns, and layers, the footnotes may be out of order. To renumber the footnotes:

1. From the Format menu choose Renumber Footnotes.

Related information

- [Table properties: notes](#)
- [Footnotes and captions](#)
- [Adding footnotes and captions](#)
- [To hide or show a caption](#)
- [To hide or show a footnote in a table](#)
- [Footnote marker](#)
- [Editing footnotes in legacy tables](#)
- [Footnote font and color settings](#)

Editing footnotes in legacy tables

For legacy tables, you can use the Edit Footnotes dialog to enter and modify footnote text and font settings, change footnote markers, and selectively hide or delete footnotes.

When you insert a new footnote in a legacy table, the Edit Footnotes dialog automatically opens. To use the Edit Footnotes dialog to edit existing footnotes (without creating a new footnote):

Marker. By default, standard footnote markers are sequential letters or numbers, depending on the table properties settings. To assign a special marker, simply enter the new marker value in the Marker column. Special markers are not affected when you renumber footnotes or switch between numbers and letters for standard markers. The display of numbers or letters for standard markers and the subscript or superscript position of footnote markers are controlled by the Footnotes tab of the Table Properties dialog. See the topic [Table properties: notes](#) for more information.

To change a special marker back to a standard marker, right-click on the marker in the Edit Footnotes dialog, select Footnote Marker from the pop-up menu, and select Standard marker in the Footnote Marker dialog box.

Footnote. The content of the footnote. The display reflects the current font and background settings. The font settings can be changed for individual footnotes using the Format subdialog. See the topic [Footnote font and color settings](#) for more information. A single background color is applied to all footnotes and can be changed in the Font and Background tab of the Cell Properties dialog. See the topic [Cell properties: Font and background](#) for more information.

Visible. All footnotes are visible by default. Deselect (clear) the Visible checkbox to hide a footnote.

- [Footnote font and color settings](#)

Related information

- [Table properties: notes](#)
- [Footnotes and captions](#)
- [Adding footnotes and captions](#)
- [To hide or show a caption](#)
- [To hide or show a footnote in a table](#)
- [Footnote marker](#)
- [Renumbering footnotes](#)
- [Footnote font and color settings](#)

Footnote font and color settings

For legacy tables, you can use the Format dialog to change the font family, style, size and color for one or more selected footnotes:

1. Double-click the footnote area of the table or from the menus choose: Format > Edit Footnotes.
2. In the Edit Footnotes dialog, select (click) one or more footnotes in the Footnotes grid.
3. Click the **Format** button.

The selected font family, style, size, and colors are applied to all the selected footnotes.

Background color, alignment, and margins can be set in the Cell Properties dialog and apply to all footnotes. You cannot change these settings for individual footnotes. See the topic [Cell properties: Font and background](#) for more information.

Related information

- [Table properties: notes](#)
- [Footnotes and captions](#)
- [Adding footnotes and captions](#)
- [To hide or show a caption](#)
- [To hide or show a footnote in a table](#)
- [Footnote marker](#)
- [Renumbering footnotes](#)
- [Editing footnotes in legacy tables](#)

Data cell widths

Set Data Cell Width is used to set all data cells to the same width.

To set the width for all data cells:

1. From the menus choose:
Format > Set Data Cell Widths...
2. Enter a value for the cell width.

Related information

- [Pivot tables](#)
 - [Manipulating a pivot table](#)
 - [Working with layers](#)
 - [Showing and hiding items](#)
 - [TableLooks](#)
 - [Table properties](#)
 - [Cell properties](#)
 - [Footnotes and captions](#)
 - [Changing column width](#)
 - [Selecting rows, columns, and cells in a pivot table](#)
 - [Printing pivot tables](#)
-

Changing column width

1. Click and drag the column border.

To view hidden borders in a pivot table, from the View menu choose Gridlines.

Related information

- [Pivot tables](#)
 - [Manipulating a pivot table](#)
 - [Working with layers](#)
 - [Showing and hiding items](#)
 - [TableLooks](#)
 - [Table properties](#)
 - [Cell properties](#)
 - [Footnotes and captions](#)
 - [Data cell widths](#)
 - [Selecting rows, columns, and cells in a pivot table](#)
 - [Printing pivot tables](#)
-

Displaying hidden borders in a pivot table

For tables without many visible borders, you can display the hidden borders. This can simplify tasks like changing column widths.

1. Activate the pivot table.
 2. From the View menu choose Gridlines.
-

Selecting rows, columns, and cells in a pivot table

You can select an entire row or column or a specified set of data and label cells.

To select multiple cells:

Select > Data and Label Cells

Related information

- [Pivot tables](#)
 - [Manipulating a pivot table](#)
 - [Working with layers](#)
 - [Showing and hiding items](#)
 - [TableLooks](#)
 - [Table properties](#)
 - [Cell properties](#)
 - [Footnotes and captions](#)
 - [Data cell widths](#)
 - [Changing column width](#)
 - [Printing pivot tables](#)
-

Printing pivot tables

Several factors can affect the way that printed pivot tables look, and these factors can be controlled by changing pivot table attributes.

- For multidimensional pivot tables (tables with layers), you can either print all layers or print only the top (visible) layer. See the topic [Table properties: printing](#) for more information.
- For long or wide pivot tables, you can automatically resize the table to fit the page or control the location of table breaks and page breaks. See the topic [Table properties: printing](#) for more information.

Use Print Preview on the File menu to see how printed pivot tables will look.

- [Controlling table breaks for wide and long tables](#)

Related information

- [Pivot tables](#)
 - [Manipulating a pivot table](#)
 - [Working with layers](#)
 - [Showing and hiding items](#)
 - [TableLooks](#)
 - [Table properties](#)
 - [Cell properties](#)
 - [Footnotes and captions](#)
 - [Data cell widths](#)
 - [Changing column width](#)
 - [Selecting rows, columns, and cells in a pivot table](#)
 - [Table properties: printing](#)
 - [Controlling table breaks for wide and long tables](#)
-

Controlling table breaks for wide and long tables

Pivot tables that are either too wide or too long to print within the defined page size are automatically split and printed in multiple sections. You can:

- Control the row and column locations where large tables are split.
- Specify rows and columns that should be kept together when tables are split.
- Rescale large tables to fit the defined page size.

To specify row and column breaks for printed pivot tables:

1. Activate the pivot table.
 2. Click any cell in the column to the left of where you want to insert the break, or click any cell in the row before the row where you want to insert the break.
 3. From the menus, choose:
`Format > Breakpoints > Vertical Breakpoint`
or
`Format > Breakpoints > Horizontal Breakpoint`
1. Activate the pivot table.
 2. Click any cell in the column to the left of where you want to insert the break, or click any cell in the row before the row where you want to insert the break.

- From the menus, choose:
Format > Breakpoints > Vertical Breakpoint

or

- Format > Breakpoints > Horizontal Breakpoint

To specify rows or columns to keep together:

- Select the labels of the rows or columns that you want to keep together. Click and drag or Shift+click to select multiple row or column labels.
- From the menus, choose:
Format > Breakpoints > Keep Together

To view breakpoints and keep together groups:

- From the menus, choose:

Format > Breakpoints > Display Breakpoints

Breakpoints are shown as vertical or horizontal lines. Keep together groups appear as grayed out rectangular regions that are enclosed by a darker border.

Note: Displaying breakpoints and keep together groups is not supported for legacy tables.

To clear breakpoints and keep together groups

To clear a breakpoint:

- Click any cell in the column to the left of a vertical breakpoint, or click any cell in the row above a horizontal breakpoint.
- From the menus, choose:
Format > Breakpoints > Clear Breakpoint or Group

To clear a keep together group:

- Select the column or row labels that specify the group.
- From the menus, choose:
Format > Breakpoints > Clear Breakpoint or Group

All breakpoints and keep together groups are automatically cleared when you pivot or reorder any row or column. This behavior does not apply to legacy tables.

Related information

- [Table properties: printing](#)
 - [Printing pivot tables](#)
-

Creating a chart from a pivot table

- Double-click the pivot table to activate it.
 - Select the rows, columns, or cells you want to display in the chart.
 - Right-click anywhere in the selected area.
 - Choose Create Graph from the pop-up menu and select a chart type.
-

Legacy tables

You can choose to render tables as legacy tables (referred to as full-featured tables in release 19) which are then fully compatible with IBM® SPSS® Statistics releases prior to 20. Legacy tables may render slowly and are only recommended if compatibility with releases prior to 20 is required. For information on how to create legacy tables, see [Pivot table options](#).

Options

- [General options](#)

- [Viewer options](#)
 - [Pivot table options](#)
 - [Output options](#)
-

General options

Maximum Number of Threads

The number of threads that multithreaded procedures use when calculating results. The Automatic setting is based on the number of available processing cores. Specify a lower value if you want to make more processing resources available to other applications while multithreaded procedures are running. This option is disabled in distributed analysis mode.

Output

Display a leading zero for decimal values. Displays leading zeros for numeric values that consist only of a decimal part. For example, when leading zeros are displayed, the value .123 is displayed as 0.123. This setting does not apply to numeric values that have a currency or percent format. Except for fixed ASCII files (*.dat), leading zeros are not included when the data are saved to an external file.

Measurement System. The measurement system used (points, inches, or centimeters) for specifying attributes such as pivot table cell margins, cell widths, and space between tables for printing.

- [Setting general options](#)
-

Setting general options

1. From the menus choose:
`Edit > Options...`
 2. Click the General tab.
 3. Select the settings that you want.
 4. Click OK or Apply.
-

Viewer options

Viewer output display options affect only new output that is produced after you change the settings. Output that is already displayed in the Viewer is not affected by changes in these settings.

Initial Output State

Controls which items are automatically displayed or hidden each time that you run a procedure and how items are initially aligned. You can control the display of the following items: log, warnings, notes, titles, pivot tables, charts, tree diagrams, and text output. You can also turn the display of commands in the log on or off. You can copy command syntax from the log and save it in a syntax file.

Note: All output items are displayed left-aligned in the Viewer. Only the alignment of printed output is affected by the justification settings. Centered and right-aligned items are identified by a small symbol.

Title

Controls the font style, size, and color for new output titles. The font Size list provides a set of predefined sizes, but you can manually enter other, supported size values.

Page Title

Controls the font style, size, and color for new page titles and page titles that are generated by **TITLE** and **SUBTITLE** command syntax or created by New Page Title on the Insert menu. The font Size list provides a set of predefined sizes, but you can manually enter other, supported size values.

Text Output

Font that is used for text output. Text output is designed for use with a monospaced (fixed-pitch) font. If you select a proportional font, tabular output does not align properly. The font Size list provides a set of predefined sizes, but you can manually enter other, supported size values.

Default Page Setup

Controls the default options for orientation and margins for printing.

- [Setting Viewer options](#)
- [Controlling the initial display state for new output](#)
- [Changing the initial alignment of new results](#)

Setting Viewer options

1. From the menus choose:
Edit>Options...
2. Click the Viewer tab.
3. Select the settings that you want.
4. Click OK or Apply.

Viewer output display options affect only new output produced after you change the settings. Output already displayed in the Viewer is not affected by changes in these settings.

Controlling the initial display state for new output

1. From the menus choose:
Edit>Options...
2. Click the Viewer tab in the Options dialog box.
3. In the Initial Output State group, select the icon for the type of item (for example, pivot table, title, notes, log) that you want to show or hide whenever new output is generated.
4. Select Shown or Hidden to specify the initial output state for that item.

Related information

- [Showing and hiding results](#)
- [To hide tables and charts](#)
- [To hide procedure results](#)

Changing the initial alignment of new results

1. From the menus choose:
Edit>Options...
2. Click the Viewer tab and select the alignment (justification) in the Initial Output State group.

Pivot table options

Pivot Table options set various options for the display of pivot tables.

TableLook

Select a TableLook from the list of files and click OK or Apply. You can use one of the TableLooks provided with IBM® SPSS® Statistics, or you can create your own in the Pivot Table Editor (choose TableLooks from the Format menu).

- Browse. Allows you to select a TableLook from another directory.
- Set TableLook Directory. Allows you to change the default TableLook directory. Use Browse to navigate to the directory you want to use, select a TableLook in that directory, and then select Set TableLook Directory.

Note: TableLooks created in earlier versions of IBM SPSS Statistics cannot be used in version 16.0 or later.

Column Widths

These options control the automatic adjustment of column widths in pivot tables.

- Adjust for labels only. Adjusts column width to the width of the column label. This produces more compact tables, but data values wider than the label may be truncated.
- Adjust for labels and data for all tables. Adjusts column width to whichever is larger: the column label or the largest data value. This produces wider tables, but it ensures that all values will be displayed.

Default Editing Mode

This option controls activation of pivot tables in the Viewer window or in a separate window. By default, double-clicking a pivot table activates all but very large tables in the Viewer window. You can choose to activate pivot tables in a separate window or select a size setting that will open smaller pivot tables in the Viewer window and larger pivot tables in a separate window.

Copying wide tables to the clipboard in rich text format

When pivot tables are pasted in Word/RTF format, tables that are too wide for the document width will either be wrapped, scaled down to fit the document width, or left unchanged.

- [Setting the TableLook for new pivot tables](#)
 - [Controlling column width for new pivot tables](#)
 - [Creating a custom default TableLook](#)
-

Setting the TableLook for new pivot tables

1. From the menus choose:
Edit > Options...
2. Click the Pivot Tables tab.
3. Select the TableLook that you want.
4. Click OK or Apply.

Note: TableLooks created in earlier versions of IBM® SPSS® Statistics cannot be used in version 16.0 or later.

Controlling column width for new pivot tables

1. From the menus choose:
Edit > Options...
 2. Click the Pivot Tables tab.
 3. Select the option you want for Column Widths.
 4. Click OK or Apply.
- **Adjust for labels and data except for extremely large tables.** For tables that don't exceed 10,000 cells, adjusts column width to whichever is larger—the column label or the largest data value. For tables with more than 10,000 cells, adjusts column width to the width of the column label.
 - **Adjust for labels only.** Adjusts column width to the width of the column label. This produces more compact tables, but data values wider than the label may be truncated.
 - **Adjust for labels and data.** Adjusts column width to whichever is larger: the column label or the largest data value. This produces wider tables, but it ensures that all values will be displayed.

Related information

- [Pivot table options](#)
 - [Setting the TableLook for new pivot tables](#)
-

Creating a custom default TableLook

1. Activate a pivot table (double-click anywhere in the table).
2. From the menus choose:
Format > TableLooks...
3. Select a TableLook from the list and click Edit Look.
4. Adjust the table properties for the attributes that you want.
5. Click Save Look or Save As to save the TableLook and click OK.
6. Deselect the pivot table.
7. From the menus choose:
Edit > Options...

8. Click the Pivot Tables tab.
9. Select the TableLook from the list and click OK.

Note: TableLooks created in earlier versions of IBM® SPSS® Statistics cannot be used in version 16.0 or later.

Output options

Output options control the default setting for a number of output options.

Screen Reader Accessibility. Controls how pivot table row and column labels are read by screen readers. You can read full row and column labels for each data cell or read only the labels that change as you move between data cells in the table.

- [Setting output options](#)
-

Setting output options

1. From the menus choose:
Edit > Options...
2. Click the Output tab.
3. Select the settings that you want.
4. Click OK or Apply.

Viewer output display options affect only new output produced after you change the settings. Output already displayed in the Viewer is not affected by changes in these settings.

Handling missing values

- [Overview of Missing Values](#)
 - [Handling Missing Values](#)
 - [Imputing or Filling Missing Values](#)
 - [CLEM Functions for Missing Values](#)
-

Overview of Missing Values

During the Data Preparation phase of data mining, you will often want to replace missing values in the data. **Missing values** are values in the data set that are unknown, uncollected, or incorrectly entered. Usually, such values are invalid for their fields. For example, the field Sex should contain the values M and F. If you discover the values Y or Z in the field, you can safely assume that such values are invalid and should therefore be interpreted as blanks. Likewise, a negative value for the field Age is meaningless and should also be interpreted as a blank. Frequently, such obviously wrong values are purposely entered, or fields left blank, during a questionnaire to indicate a nonresponse. At times, you may want to examine these blanks more closely to determine whether a nonresponse, such as the refusal to give one's age, is a factor in predicting a specific outcome.

Some modeling techniques handle missing data better than others. For example, C5.0 and Apriori cope well with values that are explicitly declared as "missing" in a Type node. Other modeling techniques have trouble dealing with missing values and experience longer training times, resulting in less-accurate models.

There are several types of missing values recognized by IBM® SPSS® Modeler:

- Null or system-missing values. These are nonstring values that have been left blank in the database or source file and have not been specifically defined as "missing" in a source or Type node. System-missing values are displayed as \$null\$. Note that empty strings are not considered nulls in IBM SPSS Modeler, although they may be treated as nulls by certain databases.
- Empty strings and white space. Empty string values and white space (strings with no visible characters) are treated as distinct from null values. Empty strings are treated as equivalent to white space for most purposes. For example, if you select the option to treat white space as blanks in a source or Type node, this setting applies to empty strings as well.
- Blank or user-defined missing values. These are values such as **unknown**, **99**, or **-1** that are explicitly defined in a source node or Type node as missing. Optionally, you can also choose to treat nulls and white space as blanks, which allows them to be flagged for special treatment and to be excluded from most calculations. For example, you can use the **@BLANK** function to treat these values, along with other types of missing values, as blanks.

Reading in mixed data. Note that when you are reading in fields with numeric storage (either integer, real, time, timestamp, or date), any non-numeric values are set to *null* or *system missing*. This is because, unlike some applications, does not allow mixed storage types within a field. To avoid this, any fields with mixed data should be read in as strings by changing the storage type in the source node or external application as necessary.

Reading empty strings from Oracle. When reading from or writing to an Oracle database, be aware that, unlike IBM SPSS Modeler and unlike most other databases, Oracle treats and stores empty string values as equivalent to null values. This means that the same data extracted from an Oracle database may behave differently than when extracted from a file or another database, and the data may return different results.

Related information

- [Handling Missing Values](#)
 - [CLEM Functions for Missing Values](#)
 - [Working with nodes](#)
-

Handling Missing Values

You should decide how to treat missing values in light of your business or domain knowledge. To ease training time and increase accuracy, you may want to remove blanks from your data set. On the other hand, the presence of blank values may lead to new business opportunities or additional insights. In choosing the best technique, you should consider the following aspects of your data:

- Size of the data set
- Number of fields containing blanks
- Amount of missing information

In general terms, there are two approaches you can follow:

- You can exclude fields or records with missing values
- You can impute, replace, or coerce missing values using a variety of methods

Both of these approaches can be largely automated using the Data Audit node. For example, you can generate a Filter node that excludes fields with too many missing values to be useful in modeling, and generate a Supernode that imputes missing values for any or all of the fields that remain. This is where the real power of the audit comes in, allowing you not only to assess the current state of your data, but to take action based on the assessment.

- [Handling Records with Missing Values](#)
- [Handling Fields with Missing Values](#)
- [Handling Records with System Missing Values](#)

Related information

- [Overview of Missing Values](#)
 - [CLEM Functions for Missing Values](#)
-

Handling Records with Missing Values

If the majority of missing values is concentrated in a small number of records, you can just exclude those records. For example, a bank usually keeps detailed and complete records on its loan customers. If, however, the bank is less restrictive in approving loans for its own staff members, data gathered for staff loans is likely to have several blank fields. In such a case, there are two options for handling these missing values:

- You can use a Select node to remove the staff records.
- If the data set is large, you can discard all records with blanks.

Related information

- [Handling Fields with Missing Values](#)
-

Handling Fields with Missing Values

If the majority of missing values is concentrated in a small number of fields, you can address them at the field level rather than at the record level. This approach also allows you to experiment with the relative importance of particular fields before deciding on an approach for handling missing values. If a field is unimportant in modeling, it probably is not worth keeping, regardless of how many missing values it has.

For example, a market research company may collect data from a general questionnaire containing 50 questions. Two of the questions address age and political persuasion, information that many people are reluctant to give. In this case, *Age* and *Political_persuasion* have many missing values.

Field Measurement Level

In determining which method to use, you should also consider the measurement level of fields with missing values.

Numeric fields. For numeric field types, such as *Continuous*, you should always eliminate any non-numeric values before building a model, because many models will not function if blanks are included in numeric fields.

Categorical fields. For categorical fields, such as *Nominal* and *Flag*, altering missing values is not necessary but will increase the accuracy of the model. For example, a model that uses the field *Sex* will still function with meaningless values, such as *Y* and *Z*, but removing all values other than *M* and *F* will increase the accuracy of the model.

Screening or Removing Fields

To screen out fields with too many missing values, you have several options:

- You can use a Data Audit node to filter fields based on quality.
- You can use a Feature Selection node to screen out fields with more than a specified percentage of missing values and to rank fields based on importance relative to a specified target.
- Instead of removing the fields, you can use a Type node to set the field role to None. This will keep the fields in the data set but exclude them from the modeling processes.

Related information

- [Handling Records with Missing Values](#)
-

Handling Records with System Missing Values

What are system missing values?

System missing values represent data values that are not known or not applicable. In databases, these values are often referred to as *NULL* values.

System missing values are different from blank values. Blank values are typically defined in the Type node as particular values, or ranges of values, which can be regarded as user-defined-missing. Blank values are handled differently in the context of modeling.

Constructing system missing values

System missing values might be present in data that is read from a data source (for example, database tables might contain *NULL* values).

System missing values can be constructed by using the value *undef* in expressions. For example, the following CLEM expression returns the *Age*, if less than or equal to 30, or a missing value if greater than 30:

```
if Age > 30 then undef else Age endif
```

Missing values can also be created when an outer join is carried out, when a number is divided by zero, when the square root of a negative number is computed, and in other situations.

Displaying system missing values

System missing values are displayed in tables and other output as `$null$`.

Testing for system missing values

Use the special function `@NULL` to return `true` if the argument value is a system missing value, for example:

```
if @NULL(MyFieldName) then 'It is null' else 'It is not null' endif
```

System missing values passed to functions

System missing values that are passed to functions usually propagate missing values to the output. For example, if the value of field `f1` is a system missing value in a particular row, then the expression `log(f1)` also evaluates to a system missing value for that row. An exception is the `@NULL` function.

System missing values in expressions that involve arithmetic operators

Applying arithmetic operators to values that include a system missing value results in a system missing value. For example, if the value of field `f1` is a system missing value in a particular row, then the expression `f1 + 10` also evaluates to a system missing value for that row.

System missing values in expressions that involve logical operators

When you work with system missing values in expressions that involve logical operators, the rules of three valued logic (*true*, *false*, and *missing*) apply and can be described in truth tables. The truth tables for the common logical operators of *not*, *and*, and *or* are shown in the following tables.

Table 1. Truth table for NOT

Operand	NOT Operand
true	false
false	true
missing	missing

Table 2. Truth table for AND

Operand1	Operand2	Operand1 AND Operand2
true	true	true
true	false	false
true	missing	missing
false	true	false
false	false	false
false	missing	false
missing	true	missing
missing	false	false
missing	missing	missing

Table 3. Truth table for OR

Operand1	Operand2	Operand1 OR Operand2
true	true	true
true	false	true
true	missing	true
false	true	true
false	false	false
false	missing	missing
missing	true	true
missing	false	missing
missing	missing	missing

System missing values in expressions that involve comparison operators

When you compare a system missing value and a non-system-missing value, the outcome evaluates to a system missing value rather than a true or false result. System missing values can be compared with each other; two system missing values are considered to be equal.

System missing values in if/then/else/endif expressions

When you use conditional expressions, and the conditional expression returns a system missing value, the value from the `else` clause is returned from the conditional expression.

System missing values in the Select node

When, for a particular record, the selection expression evaluates to a missing value, the record is not output from the Select node (this action applies to both Include and Discard modes).

System missing values in the Merge node

When you merge by using a key, any records that have system missing values in a key field are not merged.

System missing values in aggregation

When aggregating data on columns, missing values are not included in the calculation. For example, in a column with three values { 1, 2, and `undef` }, the sum of the values in the column is computed as 3; the mean value is computed as 1.5.

Related information

- [Handling Fields with Missing Values](#)
- [Handling Records with Missing Values](#)

Imputing or Filling Missing Values

In cases where there are only a few missing values, it may be useful to insert values to replace the blanks. You can do this from the Data Audit report, which allows you to specify options for specific fields as appropriate and then generate a SuperNode that imputes values using a number of methods. This is the most flexible method, and it also allows you to specify handling for large numbers of fields in a single node.

The following methods are available for imputing missing values:

Fixed. Substitutes a fixed value (either the field mean, midpoint of the range, or a constant that you specify).

Random. Substitutes a random value based on a normal or uniform distribution.

Expression. Allows you to specify a custom expression. For example, you could replace values with a global variable created by the Set Globals node.

Algorithm. Substitutes a value predicted by a model based on the C&RT algorithm. For each field imputed using this method, there will be a separate C&RT model, along with a Filler node that replaces blanks and nulls with the value predicted by the model. A Filter node is then used to remove the prediction fields generated by the model.

Alternatively, to coerce values for specific fields, you can use a Type node to ensure that the field types cover only legal values and then set the *Check* column to Coerce for the fields whose blank values need replacing.

CLEM Functions for Missing Values

There are several functions used to handle missing values. The following functions are often used in Select and Filler nodes to discard or fill missing values:

- `count_nulls(LIST)`
- `@BLANK(FIELD)`
- `@NULL(FIELD)`
- `undef`

The @ functions can be used in conjunction with the `@FIELD` function to identify the presence of blank or null values in one or more fields. The fields can simply be flagged when blank or null values are present, or they can be filled with replacement values or used in a variety of other operations.

You can count nulls across a list of fields, as follows:

```
count_nulls(['cardtenure' 'card2tenure' 'card3tenure'])
```

When using any of the functions that accept a list of fields as input, the special functions `@FIELDS_BETWEEN` and `@FIELDS_MATCHING` can be used, as shown in the following example:

```
count_nulls(@FIELDS_MATCHING('card*'))
```

You can use the `undef` function to fill fields with the system-missing value, displayed as `$null$`. For example, to replace any numeric value, you could use a conditional statement, such as:

```
if not(Age > 17) or not(Age < 66) then undef else Age endif
```

This replaces anything that is not in the range with a system-missing value, displayed as `$null$`. By using the `not()` function, you can catch all other numeric values, including any negatives. See the topic [Functions Handling Blanks and Null Values](#) for more information.

Note on Discarding Records

When using a Select node to discard records, note that syntax uses three-valued logic and automatically includes null values in select statements. To exclude null values (system-missing) in a select expression, you must explicitly specify this by using **and** **not** in the expression. For example, to select and include all records where the type of prescription drug is Drug C, you would use the following select statement:

```
Drug = 'drugC' and not (@NULL(Drug))
```

Earlier versions of excluded null values in such situations.

Related information

- [Overview of Missing Values](#)
 - [Handling Missing Values](#)
-

Building CLEM expressions

- [About CLEM](#)
 - [CLEM Examples](#)
 - [Values and Data Types](#)
 - [Expressions and Conditions](#)
 - [Stream, Session, and SuperNode Parameters](#)
 - [Working with Strings](#)
 - [Handling Blanks and Missing Values](#)
 - [Working with Numbers](#)
 - [Working with Times and Dates](#)
 - [Summarizing Multiple Fields](#)
 - [Working with Multiple-Response Data](#)
 - [The Expression Builder](#)
 - [Find and Replace](#)
-

About CLEM

The Control Language for Expression Manipulation (CLEM) is a powerful language for analyzing and manipulating the data that flows along IBM® SPSS® Modeler streams. Data miners use CLEM extensively in stream operations to perform tasks as simple as deriving profit from cost and revenue data or as complex as transforming web log data into a set of fields and records with usable information.

CLEM is used within IBM SPSS Modeler to:

- Compare and evaluate conditions on record fields.
- Derive values for new fields.
- Derive new values for existing fields.
- Reason about the sequence of records.
- Insert data from records into reports.

CLEM expressions are indispensable for data preparation in IBM SPSS Modeler and can be used in a wide range of nodes—from record and field operations (Select, Balance, Filler) to plots and output (Analysis, Report, Table). For example, you can use CLEM in a Derive node to create a new field based on a formula such as ratio.

CLEM expressions can also be used for global search and replace operations. For example, the expression `@NULL(@FIELD)` can be used in a Filler node to replace **system-missing values** with the integer value 0. (To replace **user-missing values**, also called blanks, use the `@BLANK` function.)

Related information

- [CLEM Reference Overview](#)
 - [Values and Data Types](#)
 - [Expressions and Conditions](#)
 - [The Expression Builder](#)
 - [Functions reference](#)
-

CLEM Examples

To illustrate correct syntax as well as the types of expressions possible with CLEM, example expressions follow.

Simple Expressions

Formulas can be as simple as this one, which derives a new field based on the values of the fields *After* and *Before*:

```
(After - Before) / Before * 100.0
```

Notice that field names are unquoted when referring to the values of the field.

Similarly, the following expression simply returns the log of each value for the field *salary*.

```
log(salary)
```

Complex Expressions

Expressions can also be lengthy and more complex. The following expression returns *true* if the value of two fields (*\$KX-Kohonen* and *\$KY-Kohonen*) fall within the specified ranges. Notice that here the field names are single-quoted because the field names contain special characters.

```
('$KX-Kohonen' >= -0.2635771036148072 and '$KX-Kohonen' <= 0.3146203637123107  
and '$KY-Kohonen' >= -0.18975617885589602 and  
'$KY-Kohonen' <= 0.17674794197082522) -> T
```

Several functions, such as string functions, require you to enter several parameters using correct syntax. In the following example, the function **subscr**s is used to return the first character of a *produce_ID* field, indicating whether an item is organic, genetically modified, or conventional. The results of an expression are described by -> `result`.

```
subscr(1,produce_ID) -> `c`
```

Similarly, the following expression is:

```
stripchar(`3`,`123`) -> `12`
```

It is important to note that characters are always encapsulated within single backquotes.

Combining Functions in an Expression

Frequently, CLEM expressions consist of a combination of functions. The following function combines **subscr** and **lowertoupper** to return the first character of *produce_ID* and convert it to upper case.

```
lowertoupper(subscr(1,produce_ID)) -> `C`
```

This same expression can be written in shorthand as:

```
lowertoupper(produce_ID(1)) -> `C`
```

Another commonly used combination of functions is:

```
locchar_back(`n`, (length(web_page)), web_page)
```

This expression locates the character `n` within the values of the field *web_page* reading backward from the last character of the field value. By including the **length** function as well, the expression dynamically calculates the length of the current value rather than using a static number, such as 7, which will be invalid for values with less than seven characters.

Special Functions

Numerous special functions (preceded with an @ symbol) are available. Commonly used functions include:

```
@BLANK('referrer ID') -> T
```

Frequently, special functions are used in combination, which is a commonly used method of flagging blanks in more than one field at a time.

```
@BLANK(@FIELD) -> T
```

Additional examples are discussed throughout the CLEM documentation. See the topic [CLEM Reference Overview](#) for more information.

Values and Data Types

CLEM expressions are similar to formulas constructed from values, field names, operators, and functions. The simplest valid CLEM expression is a value or a field name. Examples of valid values are:

```
3  
1.79  
'banana'
```

Examples of field names are:

```
Product_ID  
'$P-NextField'
```

where *Product* is the name of a field from a market basket data set, '*\$P-NextField*' is the name of a parameter, and the value of the expression is the value of the named field. Typically, field names start with a letter and may also contain digits and underscores (_). You can use names that do not follow these rules if you place the name within quotation marks. CLEM values can be any of the following:

- Strings--for example, "c1", "Type 2", "a piece of free text"
- Integers--for example, 12, 0, -189
- Real numbers--for example, 12.34, 0.0, -0.0045
- Date/time fields--for example, 05/12/2002, 12/05/2002, 12/05/02

It is also possible to use the following elements:

- Character codes--for example, `a` or 3
- Lists of items--for example, [1 2 3], ['Type 1'
'Type 2']

Character codes and lists do not usually occur as field values. Typically, they are used as arguments of CLEM functions.

Quoting Rules

Although the software is flexible when determining the fields, values, parameters, and strings used in a CLEM expression, the following general rules provide a list of "best practices" to use when creating expressions:

- **Strings**--Always use double quotes when writing strings ("Type 2" or "value"). Single quotes can be used instead but at the risk of confusion with quoted fields.
- **Characters**--Always use single backquotes like this `. For example, note the character d in the function `stripchar('d', "drugA")`. The only exception to this is when you are using an integer to refer to a specific character in a string. For example, note the character 5 in the function `lowertoupper("drugA"(5))`
→ "A". Note: On a standard U.K. and U.S. keyboard, the key for the backquote character (grave accent, Unicode 0060) can be found just below the Esc key.
- **Fields**--Fields are typically unquoted when used in CLEM expressions (`subscr(2, arrayID)`) → CHAR). You can use single quotes when necessary to enclose spaces or other special characters ('Order Number'). Fields that are quoted but undefined in the data set will be misread as strings.
- **Parameters**--Always use single quotes ('\$P-threshold').

Related information

- [About CLEM](#)
- [The Expression Builder](#)

Expressions and Conditions

CLEM expressions can return a result (used when deriving new values)--for example:

```
Weight * 2.2  
Age + 1  
sqrt(Signal-Echo)
```

Or, they can evaluate *true* or *false* (used when selecting on a condition)--for example:

```
Drug = "drugA"  
Age < 16  
not(PowerFlux) and Power > 2000
```

You can combine operators and functions arbitrarily in CLEM expressions--for example:

```
sqrt(abs(Signal)) * max(T1, T2) + Baseline
```

Brackets and operator precedence determine the order in which the expression is evaluated. In this example, the order of evaluation is:

- `abs(Signal)` is evaluated, and `sqrt` is applied to its result.
- `max(T1, T2)` is evaluated.
- The two results are multiplied: `*` has higher precedence than `+`.
- Finally, `Baseline` is added to the result.

The descending order of precedence (that is, operations that are performed first to operations that are performed last) is as follows:

- Function arguments
- Function calls

- xx
- x / mod div rem
- + -
- > < >= <= /== == = /=

If you want to override precedence, or if you are in any doubt of the order of evaluation, you can use parentheses to make it explicit--for example,

```
sqrt(abs(Signal)) * (max(T1, T2) + Baseline)
```

Related information

- [About CLEM](#)
- [The Expression Builder](#)

Stream, Session, and SuperNode Parameters

Parameters can be defined for use in CLEM expressions and in scripting. They are, in effect, user-defined variables that are saved and persisted with the current stream, session, or SuperNode and can be accessed from the user interface as well as through scripting. If you save a stream, for example, any parameters set for that stream are also saved. (This distinguishes them from local script variables, which can be used only in the script in which they are declared.) Parameters are often used in scripting to control the behavior of the script, by providing information about fields and values that do not need to be hard coded in the script.

The scope of a parameter depends on where it is set:

- Stream parameters can be set in a stream script or in the stream properties dialog box, and they are available to all nodes in the stream. They are displayed on the Parameters list in the Expression Builder.
- Session parameters can be set in a stand-alone script or in the session parameters dialog box. They are available to all streams used in the current session (all streams listed on the Streams tab in the managers pane).

Parameters can also be set for SuperNodes, in which case they are visible only to nodes encapsulated within that SuperNode.

Using Parameters in CLEM Expressions

Parameters are represented in CLEM expressions by `$P-pname`, where `pname` is the name of the parameter. When used in CLEM expressions, parameters must be placed within single quotes—for example, '`$P-scale`'.

Available parameters are easily viewed using the Expression Builder. To view current parameters:

1. In any dialog box accepting CLEM expressions, click the Expression Builder button.
2. From the Fields list, select Parameters.

You can select parameters from the list for insertion into the CLEM expression. See the topic [Selecting fields, parameters, and global variables](#) for more information.

Related information

- [CLEM Reference Overview](#)
- [CLEM Datatypes](#)
- [Strings](#)
- [Lists](#)
- [Fields](#)
- [CLEM Operators](#)
- [Conventions in Function Descriptions](#)

Working with Strings

There are a number of operations available for strings, including:

- Converting a string to upper case or lower case—`uppertolower(CHAR)`.
- Removing specified characters, such as `ID_` or `\$_`, from a string variable—`stripchar(CHAR, STRING)`.
- Determining the length (number of characters) for a string variable—`length(STRING)`.
- Checking the alphabetical ordering of string values—`alphabefore(STRING1, STRING2)`.
- Removing leading or trailing white space from values—`trim(STRING)`, `trim_start(STRING)`, or `trimend(STRING)`.
- Extract the first or last *n* characters from a string—`startstring(LENGTH, STRING)` or `endstring(LENGTH, STRING)`. For example, suppose you have a field named *item* that combines a product name with a four-digit ID code (`ACME CAMERA-D109`). To create a new field

that contains only the four-digit code, specify the following formula in a Derive node:

```
endstring(4, item)
```

- Matching a specific pattern—**STRING matches PATTERN**. For example, to select persons with "market" anywhere in their job title, you could specify the following in a Select node:

```
job_title matches "*market*"
```

- Replacing all instances of a substring within a string—**replace(SUBSTRING, NEWSUBSTRING, STRING)**. For example, to replace all instances of an unsupported character, such as a vertical pipe (|), with a semicolon prior to text mining, use the **replace** function in a Filler node. Under Fill in fields:, select all fields where the character may occur. For the Replace: condition, select Always, and specify the following condition under Replace with:

```
replace(' | ', ';' ,@FIELD)
```

- Deriving a flag field based on the presence of a specific substring. For example, you could use a string function in a Derive node to generate a separate flag field for each response with an expression such as:

```
hassubstring(museums, "museum_of_design")
```

See the topic [String Functions](#) for more information.

Related information

- [Handling Blanks and Missing Values](#)
- [Working with Numbers](#)
- [Working with Times and Dates](#)

Handling Blanks and Missing Values

Replacing blanks or missing values is a common data preparation task for data miners. CLEM provides you with a number of tools to automate blank handling. The Filler node is the most common place to work with blanks; however, the following functions can be used in any node that accepts CLEM expressions:

- @BLANK(FIELD)** can be used to determine records whose values are blank for a particular field, such as **Age**.
- @NULL(FIELD)** can be used to determine records whose values are system-missing for the specified field(s). In IBM® SPSS® Modeler, system-missing values are displayed as \$null\$ values.

See the topic [Functions Handling Blanks and Null Values](#) for more information.

Related information

- [CLEM Reference Overview](#)
- [Working with Strings](#)
- [Working with Numbers](#)
- [Working with Times and Dates](#)

Working with Numbers

Numerous standard operations on numeric values are available in IBM® SPSS® Modeler, such as:

- Calculating the sine of the specified angle—**sin(NUM)**
- Calculating the natural log of numeric fields—**log(NUM)**
- Calculating the sum of two numbers—**NUM1 + NUM2**

See the topic [Numeric Functions](#) for more information.

Related information

- [CLEM Reference Overview](#)
- [Working with Strings](#)
- [Handling Blanks and Missing Values](#)
- [Working with Times and Dates](#)

Working with Times and Dates

Time and date formats may vary depending on your data source and locale. The formats of date and time are specific to each stream and are set in the stream properties dialog box. The following examples are commonly used functions for working with date/time fields.

Calculating Time Passed

You can easily calculate the time passed from a baseline date using a family of functions similar to the following one. This function returns the time in months from the baseline date to the date represented by the date string **DATE** as a real number. This is an approximate figure, based on a month of 30.0 days.

```
date_in_months(Date)
```

Comparing Date/Time Values

Values of date/time fields can be compared across records using functions similar to the following one. This function returns a value of *true* if the date string **DATE1** represents a date prior to that represented by the date string **DATE2**. Otherwise, this function returns a value of 0.

```
date_before(Date1, Date2)
```

Calculating Differences

You can also calculate the difference between two times and two dates using functions, such as:

```
date_weeks_difference(Date1, Date2)
```

This function returns the time in weeks from the date represented by the date string **DATE1** to the date represented by the date string **DATE2** as a real number. This is based on a week of 7.0 days. If **DATE2** is prior to **DATE1**, this function returns a negative number.

Today's Date

The current date can be added to the data set using the function **@TODAY**. Today's date is added as a string to the specified field or new field using the date format selected in the stream properties dialog box. See the topic [Date and Time Functions](#) for more information.

Related information

- [CLEM Reference Overview](#)
- [Working with Strings](#)
- [Handling Blanks and Missing Values](#)
- [Working with Numbers](#)

Summarizing Multiple Fields

The CLEM language includes a number of functions that return summary statistics across multiple fields. These functions may be particularly useful in analyzing survey data, where multiple responses to a question may be stored in multiple fields. See the topic [Working with Multiple-Response Data](#) for more information.

Comparison Functions

You can compare values across multiple fields using the **min_n** and **max_n** functions—for example:

```
max_n(['card1fee' 'card2fee' 'card3fee' 'card4fee'])
```

You can also use a number of counting functions to obtain counts of values that meet specific criteria, even when those values are stored in multiple fields. For example, to count the number of cards that have been held for more than five years:

```
count_greater_than(5, ['cardtenure' 'card2tenure' 'card3tenure'])
```

To count null values across the same set of fields:

```
count_nulls(['cardtenure' 'card2tenure' 'card3tenure'])
```

Note that this example counts the number of cards being held, not the number of people holding them. See the topic [Comparison Functions](#) for more information.

To count the number of times a specified value occurs across multiple fields, you can use the **count_equal** function. The following example counts the number of fields in the list that contain the value **Y**.

```
count_equal("Y", [Answer1, Answer2, Answer3])
```

Given the following values for the fields in the list, the function returns the results for the value **x** as shown.

Table 1. Function values

Answer1	Answer2	Answer3	Count
Y	N	Y	2
Y	N	N	1

Numeric Functions

You can obtain statistics across multiple fields using the **sum_n**, **mean_n**, and **sdev_n** functions—for example:

```
sum_n(['card1bal' 'card2bal' 'card3bal'])  
mean_n(['card1bal' 'card2bal' 'card3bal'])
```

See the topic [Numeric Functions](#) for more information.

Generating Lists of Fields

When using any of the functions that accept a list of fields as input, the special functions **@FIELDS_BETWEEN(start, end)** and **@FIELDS_MATCHING(pattern)** can be used as input. For example, assuming the order of fields is as shown in the **sum_n** example earlier, the following would be equivalent:

```
sum_n(@FIELDS_BETWEEN(card1bal, card3bal))
```

Alternatively, to count the number of null values across all fields beginning with "card":

```
count_nulls(@FIELDS_MATCHING('card*'))
```

See the topic [Special Fields](#) for more information.

Related information

- [Functions reference](#)

Working with Multiple-Response Data

A number of comparison functions can be used to analyze multiple-response data, including:

- **value_at**
- **first_index / last_index**
- **first_non_null / last_non_null**
- **first_non_null_index / last_non_null_index**
- **min_index / max_index**

For example, suppose a multiple-response question asked for the first, second, and third most important reasons for deciding on a particular purchase (for example, price, personal recommendation, review, local supplier, other). In this case, you might determine the importance of price by deriving the index of the field in which it was first included:

```
first_index("price", [Reason1 Reason2 Reason3])
```

Similarly, suppose you have asked customers to rank three cars in order of likelihood to purchase and coded the responses in three separate fields, as follows:

Table 1. Car ranking example

customer id	car1	car2	car3
101	1	3	2
102	3	2	1
103	2	3	1

In this case, you could determine the index of the field for the car they like most (ranked #1, or the lowest rank) using the **min_index** function:

```
min_index(['car1' 'car2' 'car3'])
```

See the topic [Comparison Functions](#) for more information.

Referencing Multiple-Response Sets

The special **@MULTI_RESPONSE_SET** function can be used to reference all of the fields in a multiple-response set. For example, if the three **car** fields in the previous example are included in a multiple-response set named **car_rankings**, the following would return the same result:

```
max_index(@MULTI_RESPONSE_SET("car_rankings"))
```

Related information

- [Functions reference](#)
-

The Expression Builder

You can type CLEM expressions manually or use the Expression Builder, which displays a complete list of CLEM functions and operators as well as data fields from the current stream, allowing you to quickly build expressions without memorizing the exact names of fields or functions. In addition, the Builder controls automatically add the proper quotes for fields and values, making it easier to create syntactically correct expressions.

Note: The Expression Builder is not supported in scripting or parameter settings.

Note: If you want to change your datasource, before changing the source you should check that the Expression Builder can still support the functions you have selected. Because not all databases support all functions, you may encounter an error if you run against a new datasource.

- [Accessing the Expression Builder](#)
- [Creating Expressions](#)
- [Selecting functions](#)
- [Selecting fields, parameters, and global variables](#)
- [Viewing or selecting values](#)
- [Checking CLEM expressions](#)

Related information

- [Accessing the Expression Builder](#)
 - [Creating Expressions](#)
 - [Checking CLEM expressions](#)
 - [Find and Replace](#)
-

Accessing the Expression Builder

The Expression Builder is available in all nodes where CLEM expressions are used, including Select, Balance, Derive, Filler, Analysis, Report, and Table nodes. You can open it by clicking the calculator button just to the right of the formula field.

Related information

- [The Expression Builder](#)
 - [Creating Expressions](#)
 - [Checking CLEM expressions](#)
 - [Find and Replace](#)
-

Creating Expressions

The Expression Builder provides not only complete lists of fields, functions, and operators but also access to data values if your data is instantiated.

To Create an Expression Using the Expression Builder

1. Type in the expression field, using the function and field lists as references.
or
2. Select the required fields and functions from the scrolling lists.
3. Double-click or click the yellow arrow button to add the field or function to the expression field.
4. Use the operand buttons in the center of the dialog box to insert the operations into the expression.

Related information

- [The Expression Builder](#)
 - [Accessing the Expression Builder](#)
 - [Checking CLEM expressions](#)
 - [Find and Replace](#)
 - [Selecting functions](#)
 - [Selecting fields, parameters, and global variables](#)
 - [Viewing or selecting values](#)
-

Selecting functions

The function list displays all available CLEM functions and operators. Scroll to select a function from the list, or, for easier searching, use the drop-down list to display a subset of functions or operators. Available functions are grouped into categories for easier searching.

Most of these categories are described in the Reference section of the CLEM language description. For more information, see [Functions reference](#).

The other categories are as follows.

- General Functions contains a selection of some of the most commonly-used functions.
- Recently Used contains a list of CLEM functions used within the current session.
- @ Functions contains a list of all the special functions, which have their names preceded by an "@" sign.
Note: The `@DIFF1(FIELD1, FIELD2)` and `@DIFF2(FIELD1, FIELD2)` functions require that the two field types are the same (for example, both Integer or both Long or both Real).
- Database Functions. If the stream includes a database connection (by means of a Database source node), this selection lists the functions available from within that database, including user-defined functions (UDFs). For more information, see [Database functions](#).
- Database Aggregates. If the stream includes a database connection (by means of a Database source node), this selection lists the aggregation options available from within that database. These options are available in the Expression Builder of the Aggregate node.
- Database Window Aggregates. If the stream includes a database connection (by means of a Database source node), this selection lists the window aggregation options that you can use within that database. These options are available in the Expression Builder within nodes in the Field Operations palette only.
Note: Because SPSS® Modeler obtains the Window Aggregate Functions from the Database System View, the available options are dependent on Database behavior.
Although called "aggregates" these options are not designed for use in the Aggregate node; they are more applicable to nodes such as Derive or Select. This is because their output is scalar instead of a true aggregate; that is, they do not reduce the amount of data shown in the output in the same way that the Aggregate node does. For example, you could use this sort of aggregation to provide a moving average down through rows of data, such as "average of the current row and all previous rows".
- Built-In Aggregates. Contains a list of the possible modes of aggregation that can be used.
- Operators lists all the operators you can use when building expressions. Operators are also available from the buttons in the center of the dialog box.
- All Functions contains a complete list of available CLEM functions.

After you have selected a group of functions, double-click to insert the functions into the expression field at the point indicated by the position of the cursor.

- [Database functions](#)
-

Database functions

Database functions can be listed in many different locations; the following table shows the locations that SPSS® Modeler searches when looking for function details. This table can be used by database administrators to ensure that users have access privileges to the required areas to be able to use the different functions.

In addition, the table lists the conditions that are used to filter when a function is available for use, based on the database and function type.

Note: If using database functions from Amazon Redshift, your database administrator may need to grant you permissions to the following six database objects. The first four are system catalog tables, and the last two are schemas.

- pg_type
- pg_proc
- pg_namespace
- pg_aggregate
- information_schema
- pg_catalog

Table 1. Database functions in the Expression Builder

Database	Function type	Where to find functions	Conditions used to filter functions
Db2 LUW	UDF	SYSCAT.ROUTINES SYSCAT.ROUTINEPARMS	ROUTINETYPE is F and FUNCTIONTYPE is S
Db2 LUW	UDA	SYSCAT.ROUTINES SYSCAT.ROUTINEPARMS	ROUTINETYPE is F and FUNCTIONTYPE is C
Db2 iSeries	UDF	QSYS2.SYSROUTINES QSYS2.SYSPARMS	ROUTINE_TYPE is F and FUNCTION_TYPE is S
Db2 iSeries	UDA	QSYS2.SYSROUTINES QSYS2.SYSPARMS	ROUTINE_TYPE is F and FUNCTION_TYPE is C
Db2 z/OS	UDF	SYSIBM.SYSROUTINES SYSIBM.SYSPARMS	ROUTINETYPE is F and FUNCTIONTYPE is S
Db2 z/OS	UDA	SYSIBM.SYSROUTINES SYSIBM.SYSPARMS	ROUTINETYPE is F and FUNCTIONTYPE is C
SQL Server	UDF	SYS.ALL_OBJECTS SYS.ALL_PARAMETERS SYS.TYPES	TYPE is either FN or FS
SQL Server	UDA	SYS.ALL_OBJECTS SYS.ALL_PARAMETERS SYS.TYPES	TYPE is AF
Oracle	UDF	ALL_ARGUMENTS ALL_PROCEDURES	All of the following conditions are satisfied: <ul style="list-style-type: none"> • OBJECT_TYPE is FUNCTION • AGGREGATE is NO • PLS_TYPE is not NULL
Oracle	UDA	ALL_ARGUMENTS ALL_PROCEDURES	All of the following conditions are satisfied: <ul style="list-style-type: none"> • ARGUMENT_NAME is NULL • AGGREGATE is YES • PLS_TYPE is not NULL
Teradata	UDF	DBC.FUNCTIONS DBC.ALLRIGHTS	All of the following conditions are satisfied: <ul style="list-style-type: none"> • FUNCTIONTYPE is F • COLUMNNAME is RETURNO • SPPARAMETERTYPE is O • ACCEESSRIGHT is EF
Teradata	UDA	DBC.FUNCTIONS DBC.ALLRIGHTS	All of the following conditions are satisfied: <ul style="list-style-type: none"> • FUNCTIONTYPE is A • COLUMNNAME is RETURNO • SPPARAMETERTYPE is O • ACCEESSRIGHT is EF
Netezza	UDF	####.._V_FUNCTION NZA.._V_FUNCTION INZA.._V_FUNCTION	For ####.._V_FUNCTION, the following conditions apply: <ul style="list-style-type: none"> • RESULT does not contain a string with values such as: TABLE% • FUNCTION does not contain a string with values such as: '/_%' escape '/' • VARARGS is FALSE For both NZA.._V_FUNCTION and INZA.._V_FUNCTION, the following conditions apply: <ul style="list-style-type: none"> • RESULT does not contain a string with values such as: TABLE% • FUNCTION does not contain a string with values such as: '/_%' escape '/' • BUILTIN is f • VARARGS is FALSE
Netezza	UDA	####.._V_AGGREGATE NZA.._V_FUNCTION INZA.._V_FUNCTION	Both of the following conditions are satisfied: <ul style="list-style-type: none"> • AGGTYPE is ANY or GROUPED • VARARGS is FALSE

Database	Function type	Where to find functions	Conditions used to filter functions
Netezza	WUDA	####.._V_AGGREGATE NZA.._V_FUNCTION INZA.._V_FUNCTION	<p>For ####.._V_AGGREGATE, the following conditions apply:</p> <ul style="list-style-type: none"> • AGGTYPE is ANY or ANALYTIC • AGGREGATE is not MAX_LABEL • VARARGS is FALSE <p>For both NZA.._V_FUNCTION and INZA.._V_FUNCTION, the following conditions apply:</p> <ul style="list-style-type: none"> • AGGTYPE is ANY or ANALYTIC • BUILTIN is f • VARARGS is FALSE

Key to terms used in the table

- UDF User Defined Function
- UDA User Defined Aggregate Function
- WUDA User Defined Window Aggregate Function
- ##### the database that you are currently connected to.

Selecting fields, parameters, and global variables

The field list displays all fields available at this point in the data stream. Scroll to select a field from the list. Double-click or click the yellow arrow button to add a field to the expression.

See the topic [Stream, Session, and SuperNode Parameters](#) for more information.

In addition to fields, you can also choose from the following items:

Multiple-response sets. For more information, see the *IBM SPSS Modeler Source, Process, and Output Nodes* guide.

Recently used contains a list of fields, multiple-response sets, parameters, and global values used within the current session.

Parameters. See the topic [Stream, Session, and SuperNode Parameters](#) for more information.

Global values. For more information, see the *IBM SPSS Modeler Source, Process, and Output Nodes* guide.

Related information

- [Creating Expressions](#)
- [Selecting functions](#)
- [Viewing or selecting values](#)

Viewing or selecting values

Field values can be viewed from a number of places in the system, including the Expression Builder, data audit reports, and when editing future values in a Time Intervals node. Note that data must be fully instantiated in a source or Type node to use this feature, so that storage, types, and values are known.

To view values for a field from the Expression Builder or a Time Intervals node, select the required field and click the value picker button to open a dialog box listing values for the selected field. You can then select a value and click Insert to paste the value into the current expression or list.

Figure 1. Value picker button



To view values for a field from a Data Audit report, click on the Type cell or the Unique cell for the required field.

For flag and nominal fields, all defined values are listed. For continuous (numeric range) fields, the minimum and maximum values are displayed.

Related information

- [Creating Expressions](#)
 - [Selecting functions](#)
 - [Selecting fields, parameters, and global variables](#)
-

Checking CLEM expressions

Click Check in the Expression Builder (lower right corner) to validate the expression. Expressions that have not been checked are displayed in red. If errors are found, a message indicating the cause is displayed.

The following items are checked:

- Correct quoting of values and field names
- Correct usage of parameters and global variables
- Valid usage of operators
- Existence of referenced fields
- Existence and definition of referenced globals

If you encounter errors in syntax, try creating the expression using the lists and operator buttons rather than typing the expression manually. This method automatically adds the proper quotes for fields and values.

Note: Field names that contain separators must be surrounded by single quotes. To automatically add quotes, you can create expressions using the lists and operator buttons rather than typing expressions manually. The following characters in field names may cause errors: • ! "# \$% & ' () = ~ | - ^ ¥ @ " + * " <> ? . , / : ; → (arrow mark), □ Δ (graphic mark, etc.)

Find and Replace

The Find/Replace dialog box is available in places where you edit script or expression text, including the script editor, CLEM expression builder, or when defining a template in the Report node. When editing text in any of these areas, press **Ctrl+F** to access the dialog box, making sure cursor has focus in a text area. If working in a Filler node, for example, you can access the dialog box from any of the text areas on the Settings tab, or from the text field in the Expression Builder.

1. With the cursor in a text area, press **Ctrl+F** to access the Find/Replace dialog box.
2. Enter the text you want to search for, or choose from the drop-down list of recently searched items.
3. Enter the replacement text, if any.
4. Click Find Next to start the search.
5. Click Replace to replace the current selection, or Replace All to update all or selected instances.
6. The dialog box closes after each operation. Press **F3** from any text area to repeat the last find operation, or press **Ctrl+F** to access the dialog box again.

Search Options

Match case. Specifies whether the find operation is case-sensitive; for example, whether *myvar* matches *myVar*. Replacement text is always inserted exactly as entered, regardless of this setting.

Whole words only. Specifies whether the find operation matches text embedded within words. If selected, for example, a search on *spider* will not match *spiderman* or *spider-man*.

Regular expressions. Specifies whether regular expression syntax is used (see next section). When selected, the Whole words only option is disabled and its value is ignored.

Selected text only. Controls the scope of the search when using the Replace All option.

Regular Expression Syntax

Regular expressions allow you to search on special characters such as tabs or newline characters, classes or ranges of characters such as *a* through *d*, any digit or non-digit, and boundaries such as the beginning or end of a line. The following types of expressions are supported.

Table 1. Character matches

Characters	Matches
x	The character x
\\	The backslash character
\0n	The character with octal value 0n (0 <= n <= 7)
\0nn	The character with octal value 0nn (0 <= n <= 7)
\0mnn	The character with octal value 0mnn (0 <= m <= 3, 0 <= n <= 7)
\xhh	The character with hexadecimal value 0xhh

Characters	Matches
\uhhhh	The character with hexadecimal value 0xhhhh
\t	The tab character ('\u0009')
\n	The newline (line feed) character ('\u000A')
\r	The carriage-return character ('\u000D')
\f	The form-feed character ('\u000C')
\a	The alert (bell) character ('\u0007')
\e	The escape character ('\u001B')
\cx	The control character corresponding to x

Table 2. Matching character classes

Character classes	Matches
[abc]	a, b, or c (simple class)
[^abc]	Any character except a, b, or c (subtraction)
[a-zA-Z]	a through z or A through Z, inclusive (range)
[a-d[m-p]]	a through d, or m through p (union). Alternatively this could be specified as [a-dm-p]
[a-z&&[def]]	a through z, and d, e, or f (intersection)
[a-z&&[^bc]]	a through z, except for b and c (subtraction). Alternatively this could be specified as [ad-z]
[a-z&&[^m-p]]	a through z, and not m through p (subtraction). Alternatively this could be specified as [a-lq-z]

Table 3. Predefined character classes

Predefined character classes	Matches
.	Any character (may or may not match line terminators)
\d	Any digit: [0-9]
\D	A non-digit: [^0-9]
\s	A white space character: [\t\n\x0B\f\r]
\S	A non-white space character: [^\s]
\w	A word character: [a-zA-Z_0-9]
\W	A non-word character: [^\w]

Table 4. Boundary matches

Boundary matchers	Matches
^	The beginning of a line
\$	The end of a line
\b	A word boundary
\B	A non-word boundary
\A	The beginning of the input
\Z	The end of the input but for the final terminator, if any
\z	The end of the input

CLEM language reference

- [CLEM Reference Overview](#)
- [CLEM Datatypes](#)
- [CLEM Operators](#)
- [Functions reference](#)

CLEM Reference Overview

This section describes the Control Language for Expression Manipulation (CLEM), which is a powerful tool used to analyze and manipulate the data used in IBM® SPSS® Modeler streams. You can use CLEM within nodes to perform tasks ranging from evaluating conditions or deriving values to inserting data into reports.

CLEM expressions consist of values, field names, operators, and functions. Using the correct syntax, you can create a wide variety of powerful data operations.

Related information

- [CLEM Datatypes](#)

- [Strings](#)
 - [Lists](#)
 - [Fields](#)
 - [CLEM Operators](#)
 - [Conventions in Function Descriptions](#)
-

CLEM Datatypes

CLEM datatypes can be made up of any of the following:

- Integers
- Reals
- Characters
- Strings
- Lists
- Fields
- Date/Time

Rules for Quoting

Although IBM® SPSS® Modeler is flexible when you are determining the fields, values, parameters, and strings used in a CLEM expression, the following general rules provide a list of "good practices" to use in creating expressions:

- Strings—Always use double quotes when writing strings, such as "`Type` `2`". Single quotes can be used instead but at the risk of confusion with quoted fields.
- Fields—Use single quotes only where necessary to enclose spaces or other special characters, such as '`Order Number`'. Fields that are quoted but undefined in the data set will be misread as strings.
- Parameters—Always use single quotes when using parameters, such as '`$P-threshold`'.
- Characters—Always use single backquotes (`), such as `stripchar(`d`, "drugA")`.

These rules are covered in more detail in the following topics.

- [Integers](#)
- [Reals](#)
- [Characters](#)
- [Strings](#)
- [Lists](#)
- [Fields](#)
- [Dates](#)
- [Time](#)

Related information

- [Integers](#)
 - [Reals](#)
 - [Characters](#)
 - [Strings](#)
 - [Lists](#)
 - [Fields](#)
 - [CLEM Reference Overview](#)
 - [Stream, Session, and SuperNode Parameters](#)
 - [CLEM Operators](#)
 - [Conventions in Function Descriptions](#)
-

Integers

Integers are represented as a sequence of decimal digits. Optionally, you can place a minus sign (–) before the integer to denote a negative number—for example, `1234`, `999`, `-77`.

The CLEM language handles integers of arbitrary precision. The maximum integer size depends on your platform. If the values are too large to be displayed in an integer field, changing the field type to `Real` usually restores the value.

Related information

- [CLEM Datatypes](#)
-

Reals

Real refers to a floating-point number. Reals are represented by one or more digits followed by a decimal point followed by one or more digits. CLEM reals are held in double precision.

Optionally, you can place a minus sign (–) before the real to denote a negative number—for example, `1.234`, `0.999`, `-77.001`. Use the form `<number> e <exponent>` to express a real number in exponential notation—for example, `1234.0e5`, `1.7e-2`. When the IBM® SPSS® Modeler application reads number strings from files and converts them automatically to numbers, numbers with no leading digit before the decimal point or with no digit after the point are accepted—for example, `999.` or `.11`. However, these forms are illegal in CLEM expressions.

Note: When referencing real numbers in CLEM expressions, a period must be used as the decimal separator, regardless of any settings for the current stream or locale. For example, specify

`Na > 0.6`

rather than

`Na > 0,6`

This applies even if a comma is selected as the decimal symbol in the stream properties dialog box and is consistent with the general guideline that code syntax should be independent of any specific locale or convention.

Related information

- [CLEM Datatypes](#)
-

Characters

Characters (usually shown as **CHAR**) are typically used within a CLEM expression to perform tests on strings. For example, you can use the function **isuppercode** to determine whether the first character of a string is upper case. The following CLEM expression uses a character to indicate that the test should be performed on the first character of the string:

`isuppercode (subscrs (1, "MyString"))`

To express the code (in contrast to the location) of a particular character in a CLEM expression, use single backquotes of the form ``<character>``—for example, ``A``, ``Z``.

Note: There is no **CHAR** storage type for a field, so if a field is derived or filled with an expression that results in a **CHAR**, then that result will be converted to a string.

Related information

- [CLEM Datatypes](#)
-

Strings

Generally, you should enclose strings in double quotation marks. Examples of strings are `"c35product2"` and `"referrerID"`. To indicate special characters in a string, use a backslash—for example, `"\$65443"`. (To indicate a backslash character, use a double backslash, `\\"`.) You can use single quotes around a string, but the result is indistinguishable from a quoted field (`'referrerID'`). See the topic [String Functions](#) for more information.

Related information

- [CLEM Datatypes](#)
 - [CLEM Reference Overview](#)
 - [Lists](#)
 - [Fields](#)
 - [CLEM Operators](#)
 - [Conventions in Function Descriptions](#)
-

Lists

A list is an ordered sequence of elements, which may be of mixed type. Lists are enclosed in square brackets ([]). Examples of lists are [1 2 4 16] and ["abc" "def"]. Lists are not used as the value of IBM® SPSS® Modeler fields. They are used to provide arguments to functions, such as `member` and `oneof`.

Note: Lists can be composed only of static objects (for example, a string, number, or field name) and not calls to functions.

Related information

- [CLEM Datatypes](#)
- [CLEM Reference Overview](#)
- [Strings](#)
- [Fields](#)
- [CLEM Operators](#)
- [Conventions in Function Descriptions](#)

Fields

Names in CLEM expressions that are not names of functions are assumed to be field names. You can write these simply as `Power`, `val27`, `state_flag`, and so on, but if the name begins with a digit or includes non-alphabetic characters, such as spaces (with the exception of the underscore), place the name within single quotation marks—for example, '`Power Increase`', '`2nd answer`', '`#101`', '`$P-NextField`'.

Note: Fields that are quoted but undefined in the data set will be misread as strings.

Related information

- [CLEM Datatypes](#)
- [CLEM Reference Overview](#)
- [Strings](#)
- [Lists](#)
- [CLEM Operators](#)
- [Conventions in Function Descriptions](#)

Dates

Date calculations are based on a "baseline" date, which is specified in the stream properties dialog box. The default baseline date is 1 January 1900.

The CLEM language supports the following date formats.

Table 1. CLEM language date formats

Format	Examples
DDMMYY	150163
MMDDYY	011563
YYMMDD	630115
YYYYMMDD	19630115
YYYYDDD	Four-digit year followed by a three-digit number representing the day of the year—for example, 2000032 represents the 32nd day of 2000, or 1 February 2000.
DAY	Day of the week in the current locale—for example, <code>Monday</code> , <code>Tuesday</code> , ..., in English.
MONTH	Month in the current locale—for example, <code>January</code> , <code>February</code> ,
DD/MM/YY	15/01/63
DD/MM/YYYY	15/01/1963
MM/DD/YY	01/15/63
MM/DD/YYYY	01/15/1963
DD-MM-YY	15-01-63
DD-MM-YYYY	15-01-1963
MM-DD-YY	01-15-63
MM-DD-YYYY	01-15-1963

Format	Examples
DD.MM.YY	15.01.63
DD.MM.YYYY	15.01.1963
MM.DD.YY	01.15.63
MM.DD.YYYY	01.15.1963
DD-MON-YY	15-JAN-63, 15-jan-63, 15-Jan-63
DD/MON/YY	15/JAN/63, 15/jan/63, 15/Jan/63
DD.MON.YY	15.JAN.63, 15.jan.63, 15.Jan.63
DD-MON-YYYY	15-JAN-1963, 15-jan-1963, 15-Jan-1963
DD/MON/YYYY	15/JAN/1963, 15/jan/1963, 15/Jan/1963
DD.MON.YYYY	15.JAN.1963, 15.jan.1963, 15.Jan.1963
MON YYYY	Jan 2004
q Q YYYY	Date represented as a digit (1–4) representing the quarter followed by the letter Q and a four-digit year—for example, 25 December 2004 would be represented as 4 Q 2004.
ww WK YYYY	Two-digit number representing the week of the year followed by the letters WK and then a four-digit year. The week of the year is calculated assuming that the first day of the week is Monday and there is at least one day in the first week.

Related information

- [CLEM Reference Overview](#)

Time

The CLEM language supports the following time formats.

Table 1. CLEM language time formats

Format	Examples
HHMMSS	120112, 010101, 221212
HHMM	1223, 0745, 2207
MMSS	5558, 0100
HH:MM:SS	12:01:12, 01:01:01, 22:12:12
HH:MM	12:23, 07:45, 22:07
MM:SS	55:58, 01:00
(H) H: (M) M: (S) S	12:1:12, 1:1:1, 22:12:12
(H) H: (M) M	12:23, 7:45, 22:7
(M) M: (S) S	55:58, 1:0
HH.MM.SS	12.01.12, 01.01.01, 22.12.12
HH.MM	12.23, 07.45, 22.07
MM.SS	55.58, 01.00
(H) H. (M) M. (S) S	12.1.12, 1.1.1, 22.12.12
(H) H. (M) M	12.23, 7.45, 22.7
(M) M. (S) S	55.58, 1.0

Related information

- [CLEM Reference Overview](#)

CLEM Operators

The following operators are available.

Table 1. CLEM language operators

Operation	Comments	Precedence (see next section)
or	Used between two CLEM expressions. Returns a value of true if either is true or if both are true.	10
and	Used between two CLEM expressions. Returns a value of true if both are true.	9
=	Used between any two comparable items. Returns true if ITEM1 is equal to ITEM2.	7
==	Identical to =.	7

Operation	Comments	Precedence (see next section)
<code>/=</code>	Used between any two comparable items. Returns true if ITEM1 is <i>not</i> equal to ITEM2.	7
<code>/==</code>	Identical to <code>/=</code> .	7
<code>></code>	Used between any two comparable items. Returns true if ITEM1 is strictly greater than ITEM2.	6
<code>>=</code>	Used between any two comparable items. Returns true if ITEM1 is greater than or equal to ITEM2.	6
<code><</code>	Used between any two comparable items. Returns true if ITEM1 is strictly less than ITEM2	6
<code><=</code>	Used between any two comparable items. Returns true if ITEM1 is less than or equal to ITEM2.	6
<code>&&_0</code>	Used between two integers. Equivalent to the Boolean expression INT1 && INT2 = 0.	6
<code>&&/=_0</code>	Used between two integers. Equivalent to the Boolean expression INT1 && INT2 /= 0.	6
<code>+</code>	Adds two numbers: NUM1 + NUM2.	5
<code>>x</code>	Concatenates two strings; for example, <code>STRING1 >x STRING2</code> .	5
<code>-</code>	Subtracts one number from another: NUM1 - NUM2. Can also be used in front of a number: - NUM.	5
<code>*</code>	Used to multiply two numbers: NUM1 * NUM2.	4
<code>&&</code>	Used between two integers. The result is the bitwise 'and' of the integers INT1 and INT2.	4
<code>&&~~</code>	Used between two integers. The result is the bitwise 'and' of INT1 and the bitwise complement of INT2.	4
<code> </code>	Used between two integers. The result is the bitwise 'inclusive or' of INT1 and INT2.	4
<code>~~</code>	Used in front of an integer. Produces the bitwise complement of INT.	4
<code> /&</code>	Used between two integers. The result is the bitwise 'exclusive or' of INT1 and INT2.	4
<code>INT1 << N</code>	Used between two integers. Produces the bit pattern of INT shifted left by N positions.	4
<code>INT1 >> N</code>	Used between two integers. Produces the bit pattern of INT shifted right by N positions.	4
<code>/</code>	Used to divide one number by another: NUM1 / NUM2.	4
<code>**</code>	Used between two numbers: BASE ** POWER. Returns BASE raised to the power POWER.	3
<code>rem</code>	Used between two integers: INT1 rem INT2. Returns the remainder, INT1 - (INT1 div INT2) * INT2.	2
<code>div</code>	Used between two integers: INT1 div INT2. Performs integer division.	2

Operator Precedence

Precedences determine the parsing of complex expressions, especially unbracketed expressions with more than one infix operator. For example,

`3 + 4 * 5`

parses as `3 + (4 * 5)` rather than `(3 + 4) *`

because the relative precedences dictate that `*` is to be parsed before `+`. Every operator in the CLEM language has a precedence value associated with it; the lower this value, the more important it is on the parsing list, meaning that it will be processed sooner than other operators with higher precedence values.

Related information

- [CLEM Reference Overview](#)
- [CLEM Datatypes](#)
- [Strings](#)
- [Lists](#)
- [Fields](#)
- [Conventions in Function Descriptions](#)

Functions reference

The following CLEM functions are available for working with data in IBM® SPSS® Modeler. You can enter these functions as code in various dialog boxes, such as Derive and Set To Flag nodes, or you can use the Expression Builder to create valid CLEM expressions without memorizing function lists or field names.

Table 1. CLEM functions for use with IBM SPSS Modeler data

Function Type	Description
Information	Used to gain insight into field values. For example, the function <code>is_string</code> returns true for all records whose type is a string.
Conversion	Used to construct new fields or convert storage type. For example, the function <code>to_timestamp</code> converts the selected field to a timestamp.

Function Type	Description
Comparison	Used to compare field values to each other or to a specified string. For example, <code><=</code> is used to compare whether the values of two fields are lesser or equal.
Logical	Used to perform logical operations, such as <code>if, then, else</code> operations.
Numeric	Used to perform numeric calculations, such as the natural log of field values.
Trigonometric	Used to perform trigonometric calculations, such as the arccosine of a specified angle.
Probability	Returns probabilities that are based on various distributions, such as probability that a value from Student's <i>t</i> distribution is less than a specific value.
Spatial	Used to perform spatial calculations on geospatial data.
Bitwise	Used to manipulate integers as bit patterns.
Random	Used to randomly select items or generate numbers.
String	Used to perform various operations on strings, such as <code>stripchar</code> , which allows you to remove a specified character.
SoundEx	Used to find strings when the precise spelling is not known; based on phonetic assumptions about how certain letters are pronounced.
Date and time	Used to perform various operations on date, time, and timestamp fields.
Sequence	Used to gain insight into the record sequence of a data set or perform operations that are based on that sequence.
Global	Used to access global values that are created by a Set Globals node. For example, <code>@MEAN</code> is used to refer to the mean average of all values for a field across the entire data set.
Blanks and null	Used to access, flag, and frequently fill user-specified blanks or system-missing values. For example, <code>@BLANK (FIELD)</code> is used to raise a true flag for records where blanks are present.
Special fields	Used to denote the specific fields under examination. For example, <code>@FIELD</code> is used when deriving multiple fields.

- [Conventions in Function Descriptions](#)
- [Information Functions](#)
- [Conversion Functions](#)
- [Comparison Functions](#)
- [Logical Functions](#)
- [Numeric Functions](#)
- [Trigonometric Functions](#)
- [Probability Functions](#)
- [Spatial functions](#)
- [Bitwise Integer Operations](#)
- [Random Functions](#)
- [String Functions](#)
- [SoundEx Functions](#)
- [Date and Time Functions](#)
- [Sequence functions](#)
- [Global Functions](#)
- [Functions Handling Blanks and Null Values](#)
- [Special Fields](#)

Conventions in Function Descriptions

The following conventions are used throughout this guide when referring to items in a function.

Table 1. Conventions in function descriptions

Convention	Description
<code>BOOL</code>	A Boolean, or flag, such as true or false.
<code>NUM, NUM1, NUM2</code>	Any number.
<code>REAL, REAL1, REAL2</code>	Any real number, such as <code>1.234</code> or <code>-77.01</code> .
<code>INT, INT1, INT2</code>	Any integer, such as <code>1</code> or <code>-77</code> .
<code>CHAR</code>	A character code, such as <code>'A'</code> .
<code>STRING</code>	A string, such as <code>"referrerID"</code> .
<code>LIST</code>	A list of items, such as <code>["abc" "def"]</code> .
<code>ITEM</code>	A field, such as <code>Customer</code> or <code>extract_concept</code> .
<code>DATE</code>	A date field, such as <code>start_date</code> , where values are in a format such as <code>DD-MON-YYYY</code> .
<code>TIME</code>	A time field, such as <code>power_flux</code> , where values are in a format such as <code>HHMMSS</code> .

Functions in this guide are listed with the function in one column, the result type (integer, string, and so on) in another, and a description (where available) in a third column. For example, the following is the description of the `rem` function.

Table 2. rem function description

Function	Result	Description
<code>INT1 rem INT2</code>	Number	Returns the remainder of <code>INT1</code> divided by <code>INT2</code> . For example, <code>INT1 - (INT1 div INT2) * INT2</code> .

Details on usage conventions, such as how to list items or specify characters in a function, are described elsewhere. See the topic [CLEM Datatypes](#) for more information.

Related information

- [CLEM Datatypes](#)
 - [CLEM Reference Overview](#)
 - [Strings](#)
 - [Lists](#)
 - [Fields](#)
 - [CLEM Operators](#)
-

Information Functions

Information functions are used to gain insight into the values of a particular field. They are typically used to derive flag fields. For example, you can use the `@BLANK` function to create a flag field indicating records whose values are blank for the selected field. Similarly, you can check the storage type for a field using any of the storage type functions, such as `is_string`.

Table 1. CLEM information functions

Function	Result	Description
<code>@BLANK(FIELD)</code>	Boolean	Returns true for all records whose values are blank according to the blank-handling rules set in an upstream Type node or source node (Types tab).
<code>@NULL(ITEM)</code>	Boolean	Returns true for all records whose values are undefined. Undefined values are system null values, displayed in IBM® SPSS® Modeler as <code>\$null\$</code> .
<code>is_date(ITEM)</code>	Boolean	Returns true for all records whose type is a date.
<code>is_datetime(ITEM)</code>	Boolean	Returns true for all records whose type is a date, time, or timestamp.
<code>is_integer(ITEM)</code>	Boolean	Returns true for all records whose type is an integer.
<code>is_number(ITEM)</code>	Boolean	Returns true for all records whose type is a number.
<code>is_real(ITEM)</code>	Boolean	Returns true for all records whose type is a real.
<code>is_string(ITEM)</code>	Boolean	Returns true for all records whose type is a string.
<code>is_time(ITEM)</code>	Boolean	Returns true for all records whose type is a time.
<code>is_timestamp(ITEM)</code>	Boolean	Returns true for all records whose type is a timestamp.

Related information

- [Functions reference](#)
-

Conversion Functions

Conversion functions allow you to construct new fields and convert the storage type of existing files. For example, you can form new strings by joining strings together or by taking strings apart. To join two strings, use the operator `><`. For example, if the field `site` has the value `"BRAMLEY"`, then `"xx"` `>< site` returns `"xxBRAMLEY"`. The result of `><` is always a string, even if the arguments are not strings. Thus, if field `v1` is 3 and field `v2` is 5, then `v1 >< v2` returns `"35"` (a string, not a number).

Conversion functions (and any other functions that require a specific type of input, such as a date or time value) depend on the current formats specified in the Stream Options dialog box. For example, if you want to convert a string field with values `Jan 2003`, `Feb 2003`, and so on, select the matching date format `MON YYYY` as the default date format for the stream.

Table 1. CLEM conversion functions

Function	Result	Description
----------	--------	-------------

Function	Result	Description
<code>ITEM1 <> ITEM2</code>	<code>String</code>	Concatenates values for two fields and returns the resulting string as <code>ITEM1ITEM2</code> .
<code>to_integer(ITEM)</code>	<code>Integer</code>	Converts the storage of the specified field to an integer.
<code>to_real(ITEM)</code>	<code>Real</code>	Converts the storage of the specified field to a real.
<code>to_number(ITEM)</code>	<code>Number</code>	Converts the storage of the specified field to a number.
<code>to_string(ITEM)</code>	<code>String</code>	Converts the storage of the specified field to a string. When a real is converted to string using this function, it returns a value with 6 digits after the radix point.
<code>to_time(ITEM)</code>	<code>Time</code>	Converts the storage of the specified field to a time.
<code>to_date(ITEM)</code>	<code>Date</code>	Converts the storage of the specified field to a date.
<code>to_timestamp(ITEM)</code>	<code>Timestamp</code>	Converts the storage of the specified field to a timestamp.
<code>to_datetime(ITEM)</code>	<code>Datetime</code>	Converts the storage of the specified field to a date, time, or timestamp value.
<code>datetime_date(ITEM)</code>	<code>Date</code>	Returns the date value for a <i>number</i> , <i>string</i> , or <i>timestamp</i> . Note this is the only function that allows you to convert a number (in seconds) back to a date. If <code>ITEM</code> is a string, creates a date by parsing a string in the current date format. The date format specified in the stream properties dialog box must be correct for this function to be successful. If <code>ITEM</code> is a number, it is interpreted as a number of seconds since the base date (or epoch). Fractions of a day are truncated. If <code>ITEM</code> is a timestamp, the date part of the timestamp is returned. If <code>ITEM</code> is a date, it is returned unchanged.
<code>stb_centroid_latitude(ITEM)</code>	<code>Integer</code>	Returns an integer value for latitude corresponding to centroid of the geohash argument.
<code>stb_centroid_longitude(ITEM)</code>	<code>Integer</code>	Returns an integer value for longitude corresponding to centroid of the geohash argument.
<code>to_geohash(ITEM)</code>	<code>String</code>	Returns the geohashed string corresponding to the latitude and longitude using the specified number of bits for the density. A geohash is a code used to identify a set of geographic coordinates based on the latitude and longitude details. The three parameters for <code>to_geohash</code> are: <ul style="list-style-type: none"> • <i>latitude</i>: Range (-180, 180), and units are degrees in the WGS84 coordinate system • <i>longitude</i>: Range (-90, 90), and units are degrees in the WGS84 coordinate system • <i>bits</i>: The number of bits to use to store the hash. Range [1,75]. This affects both the length of the returned string (1 character is used for every 5 bits), and the accuracy of the hash. For example, 5 bits (1 character) represents approximately 2500 kilometers, or 45 bits (9 characters), represents approximately 2.3 meters.

Related information

- [Functions reference](#)
- [Date and Time Functions](#)
- [Converting Date and Time Values](#)

Comparison Functions

Comparison functions are used to compare field values to each other or to a specified string. For example, you can check strings for equality using `=`. An example of string equality verification is: `Class = "class 1"`.

For purposes of numeric comparison, *greater* means closer to positive infinity, and *lesser* means closer to negative infinity. That is, all negative numbers are less than any positive number.

Table 1. CLEM comparison functions

Function	Result	Description
<code>count_equal(ITEM1, LIST)</code>	<code>Integer</code>	Returns the number of values from a list of fields that are equal to <code>ITEM1</code> or null if <code>ITEM1</code> is null.
<code>count_greater_than(ITEM1, LIST)</code>	<code>Integer</code>	Returns the number of values from a list of fields that are greater than <code>ITEM1</code> or null if <code>ITEM1</code> is null.
<code>count_less_than(ITEM1, LIST)</code>	<code>Integer</code>	Returns the number of values from a list of fields that are less than <code>ITEM1</code> or null if <code>ITEM1</code> is null.

Function	Result	Description
<code>count_not_equal(ITEM1, LIST)</code>	<code>Integer</code>	Returns the number of values from a list of fields that are not equal to <i>ITEM1</i> or null if <i>ITEM1</i> is null.
<code>count_nulls(LIST)</code>	<code>Integer</code>	Returns the number of null values from a list of fields.
<code>count_non_nulls(LIST)</code>	<code>Integer</code>	Returns the number of non-null values from a list of fields.
<code>date_before(DATE1, DATE2)</code>	<code>Boolean</code>	Used to check the ordering of date values. Returns a true value if <i>DATE1</i> is before <i>DATE2</i> .
<code>first_index(ITEM, LIST)</code>	<code>Integer</code>	Returns the index of the first field containing <i>ITEM</i> from a <i>LIST</i> of fields or 0 if the value is not found. Supported for string, integer, and real types only.
<code>first_non_null(LIST)</code>	<code>Any</code>	Returns the first non-null value in the supplied list of fields. All storage types supported.
<code>first_non_null_index(LIST)</code>	<code>Integer</code>	Returns the index of the first field in the specified <i>LIST</i> containing a non-null value or 0 if all values are null. All storage types are supported.
<code>ITEM1 = ITEM2</code>	<code>Boolean</code>	Returns true for records where <i>ITEM1</i> is equal to <i>ITEM2</i> .
<code>ITEM1 /= ITEM2</code>	<code>Boolean</code>	Returns true if the two strings are not identical or 0 if they are identical.
<code>ITEM1 < ITEM2</code>	<code>Boolean</code>	Returns true for records where <i>ITEM1</i> is less than <i>ITEM2</i> .
<code>ITEM1 <= ITEM2</code>	<code>Boolean</code>	Returns true for records where <i>ITEM1</i> is less than or equal to <i>ITEM2</i> .
<code>ITEM1 > ITEM2</code>	<code>Boolean</code>	Returns true for records where <i>ITEM1</i> is greater than <i>ITEM2</i> .
<code>ITEM1 >= ITEM2</code>	<code>Boolean</code>	Returns true for records where <i>ITEM1</i> is greater than or equal to <i>ITEM2</i> .
<code>last_index(ITEM, LIST)</code>	<code>Integer</code>	Returns the index of the last field containing <i>ITEM</i> from a <i>LIST</i> of fields or 0 if the value is not found. Supported for string, integer, and real types only.
<code>last_non_null(LIST)</code>	<code>Any</code>	Returns the last non-null value in the supplied list of fields. All storage types supported.
<code>last_non_null_index(LIST)</code>	<code>Integer</code>	Returns the index of the last field in the specified <i>LIST</i> containing a non-null value or 0 if all values are null. All storage types are supported.
<code>max(ITEM1, ITEM2)</code>	<code>Any</code>	Returns the greater of the two items-- <i>ITEM1</i> or <i>ITEM2</i> .
<code>max_index(LIST)</code>	<code>Integer</code>	Returns the index of the field containing the maximum value from a list of numeric fields or 0 if all values are null. For example, if the third field listed contains the maximum, the index value 3 is returned. If multiple fields contain the maximum value, the one listed first (leftmost) is returned.
<code>max_n(LIST)</code>	<code>Number</code>	Returns the maximum value from a list of numeric fields or null if all of the field values are null.
<code>member(ITEM, LIST)</code>	<code>Boolean</code>	Returns true if <i>ITEM</i> is a member of the specified <i>LIST</i> . Otherwise, a false value is returned. A list of field names can also be specified.
<code>min(ITEM1, ITEM2)</code>	<code>Any</code>	Returns the lesser of the two items-- <i>ITEM1</i> or <i>ITEM2</i> .
<code>min_index(LIST)</code>	<code>Integer</code>	Returns the index of the field containing the minimum value from a list of numeric fields or 0 if all values are null. For example, if the third field listed contains the minimum, the index value 3 is returned. If multiple fields contain the minimum value, the one listed first (leftmost) is returned.
<code>min_n(LIST)</code>	<code>Number</code>	Returns the minimum value from a list of numeric fields or null if all of the field values are null.
<code>time_before(TIME1, TIME2)</code>	<code>Boolean</code>	Used to check the ordering of time values. Returns a true value if <i>TIME1</i> is before <i>TIME2</i> .
<code>value_at(INT, LIST)</code>		Returns the value of each listed field at offset <i>INT</i> or NULL if the offset is outside the range of valid values (that is, less than 1 or greater than the number of listed fields). All storage types supported.

Related information

- [Functions reference](#)

Logical Functions

CLEM expressions can be used to perform logical operations.

Table 1. CLEM logical functions

Function	Result	Description
----------	--------	-------------

Function	Result	Description
<code>COND1 and COND2</code>	<code>Boolean</code>	This operation is a logical conjunction and returns a true value if both <code>COND1</code> and <code>COND2</code> are true. If <code>COND1</code> is false, then <code>COND2</code> is not evaluated; this makes it possible to have conjunctions where <code>COND1</code> first tests that an operation in <code>COND2</code> is legal. For example, <code>length(Label) >=6 and Label(6) = 'x'</code> .
<code>COND1 or COND2</code>	<code>Boolean</code>	This operation is a logical (inclusive) disjunction and returns a true value if either <code>COND1</code> or <code>COND2</code> is true or if both are true. If <code>COND1</code> is true, <code>COND2</code> is not evaluated.
<code>not(COND)</code>	<code>Boolean</code>	This operation is a logical negation and returns a true value if <code>COND</code> is false. Otherwise, this operation returns a value of 0.
<code>if COND then EXPR1 else EXPR2 endif</code>	<code>Any</code>	This operation is a conditional evaluation. If <code>COND</code> is true, this operation returns the result of <code>EXPR1</code> . Otherwise, the result of evaluating <code>EXPR2</code> is returned.
<code>if COND1 then EXPR1 elseif COND2 then EXPR2 else EXPR_N endif</code>	<code>Any</code>	This operation is a multibranch conditional evaluation. If <code>COND1</code> is true, this operation returns the result of <code>EXPR1</code> . Otherwise, if <code>COND2</code> is true, this operation returns the result of evaluating <code>EXPR2</code> . Otherwise, the result of evaluating <code>EXPR_N</code> is returned.

Related information

- [Functions reference](#)

Numeric Functions

CLEM contains a number of commonly used numeric functions.

Table 1. CLEM numeric functions

Function	Result	Description
<code>-NUM</code>	<code>Number</code>	Used to negate <code>NUM</code> . Returns the corresponding number with the opposite sign.
<code>NUM1 + NUM2</code>	<code>Number</code>	Returns the sum of <code>NUM1</code> and <code>NUM2</code> .
<code>NUM1 - NUM2</code>	<code>Number</code>	Returns the value of <code>NUM2</code> subtracted from <code>NUM1</code> .
<code>NUM1 * NUM2</code>	<code>Number</code>	Returns the value of <code>NUM1</code> multiplied by <code>NUM2</code> .
<code>NUM1 / NUM2</code>	<code>Number</code>	Returns the value of <code>NUM1</code> divided by <code>NUM2</code> .
<code>INT1 div INT2</code>	<code>Number</code>	Used to perform integer division. Returns the value of <code>INT1</code> divided by <code>INT2</code> .
<code>INT1 rem INT2</code>	<code>Number</code>	Returns the remainder of <code>INT1</code> divided by <code>INT2</code> . For example, <code>INT1 - (INT1 div INT2) * INT2</code> .
<code>INT1 mod INT2</code>	<code>Number</code>	This function has been deprecated. Use the <code>rem</code> function instead.
<code>BASE ** POWER</code>	<code>Number</code>	Returns <code>BASE</code> raised to the power <code>POWER</code> , where either may be any number (except that <code>BASE</code> must not be zero if <code>POWER</code> is zero of any type other than integer 0). If <code>POWER</code> is an integer, the computation is performed by successively multiplying powers of <code>BASE</code> . Thus, if <code>BASE</code> is an integer, the result will be an integer. If <code>POWER</code> is integer 0, the result is always a 1 of the same type as <code>BASE</code> . Otherwise, if <code>POWER</code> is not an integer, the result is computed as <code>exp(POWER * log(BASE))</code> .
<code>abs(NUM)</code>	<code>Number</code>	Returns the absolute value of <code>NUM</code> , which is always a number of the same type.
<code>exp(NUM)</code>	<code>Real</code>	Returns e raised to the power <code>NUM</code> , where e is the base of natural logarithms.
<code>fracof(NUM)</code>	<code>Real</code>	Returns the fractional part of <code>NUM</code> , defined as <code>NUM - intof(NUM)</code> .
<code>intof(NUM)</code>	<code>Integer</code>	Truncates its argument to an integer. It returns the integer of the same sign as <code>NUM</code> and with the largest magnitude such that <code>abs(INT) <= abs(NUM)</code> .
<code>log(NUM)</code>	<code>Real</code>	Returns the natural (base e) logarithm of <code>NUM</code> , which must not be a zero of any kind.
<code>log10(NUM)</code>	<code>Real</code>	Returns the base 10 logarithm of <code>NUM</code> , which must not be a zero of any kind. This function is defined as <code>log(NUM) / log(10)</code> .
<code>negate(NUM)</code>	<code>Number</code>	Used to negate <code>NUM</code> . Returns the corresponding number with the opposite sign.
<code>round(NUM)</code>	<code>Integer</code>	Used to round <code>NUM</code> to an integer by taking <code>intof(NUM+0.5)</code> if <code>NUM</code> is positive or <code>intof(NUM-0.5)</code> if <code>NUM</code> is negative.
<code>sign(NUM)</code>	<code>Number</code>	Used to determine the sign of <code>NUM</code> . This operation returns -1, 0, or 1 if <code>NUM</code> is an integer. If <code>NUM</code> is a real, it returns -1.0, 0.0, or 1.0, depending on whether <code>NUM</code> is negative, zero, or positive.
<code>sqrt(NUM)</code>	<code>Real</code>	Returns the square root of <code>NUM</code> . <code>NUM</code> must be positive.
<code>sum_n(LIST)</code>	<code>Number</code>	Returns the sum of values from a list of numeric fields or null if all of the field values are null.
<code>mean_n(LIST)</code>	<code>Number</code>	Returns the mean value from a list of numeric fields or null if all of the field values are null.
<code>sdev_n(LIST)</code>	<code>Number</code>	Returns the standard deviation from a list of numeric fields or null if all of the field values are null.

Related information

- [Functions reference](#)

Trigonometric Functions

All of the functions in this section either take an angle as an argument or return one as a result. In both cases, the units of the angle (radians or degrees) are controlled by the setting of the relevant stream option.

Table 1. CLEM trigonometric functions

Function	Result	Description
<code>arccos (NUM)</code>	<i>Real</i>	Computes the arccosine of the specified angle.
<code>arccosh (NUM)</code>	<i>Real</i>	Computes the hyperbolic arccosine of the specified angle.
<code>arcsin (NUM)</code>	<i>Real</i>	Computes the arcsine of the specified angle.
<code>arcsinh (NUM)</code>	<i>Real</i>	Computes the hyperbolic arcsine of the specified angle.
<code>arctan (NUM)</code>	<i>Real</i>	Computes the arctangent of the specified angle.
<code>arctan2 (NUM_Y, NUM_X)</code>	<i>Real</i>	Computes the arctangent of <code>NUM_Y / NUM_X</code> and uses the signs of the two numbers to derive quadrant information. The result is a real in the range <code>- pi < ANGLE <= pi (radians) - 180 < ANGLE <= 180 (degrees)</code>
<code>arctanh (NUM)</code>	<i>Real</i>	Computes the hyperbolic arctangent of the specified angle.
<code>cos (NUM)</code>	<i>Real</i>	Computes the cosine of the specified angle.
<code>cosh (NUM)</code>	<i>Real</i>	Computes the hyperbolic cosine of the specified angle.
<code>pi</code>	<i>Real</i>	This constant is the best real approximation to pi.
<code>sin (NUM)</code>	<i>Real</i>	Computes the sine of the specified angle.
<code>sinh (NUM)</code>	<i>Real</i>	Computes the hyperbolic sine of the specified angle.
<code>tan (NUM)</code>	<i>Real</i>	Computes the tangent of the specified angle.
<code>tanh (NUM)</code>	<i>Real</i>	Computes the hyperbolic tangent of the specified angle.

Related information

- [Functions reference](#)

Probability Functions

Probability functions return probabilities based on various distributions, such as the probability that a value from Student's *t* distribution will be less than a specific value.

Table 1. CLEM probability functions

Function	Result	Description
<code>cdf_chisq (NUM, DF)</code>	<i>Real</i>	Returns the probability that a value from the chi-square distribution with the specified degrees of freedom will be less than the specified number.
<code>cdf_f (NUM, DF1, DF2)</code>	<i>Real</i>	Returns the probability that a value from the <i>F</i> distribution, with degrees of freedom <i>DF1</i> and <i>DF2</i> , will be less than the specified number.
<code>cdf_normal (NUM, MEAN, STDDEV)</code>	<i>Real</i>	Returns the probability that a value from the normal distribution with the specified mean and standard deviation will be less than the specified number.
<code>cdf_t (NUM, DF)</code>	<i>Real</i>	Returns the probability that a value from Student's <i>t</i> distribution with the specified degrees of freedom will be less than the specified number.

Related information

- [Functions reference](#)

Spatial functions

Spatial functions can be used with geospatial data. For example, they allow you to calculate the distances between two points, the area of a polygon, and so on. There can also be situations that require a merge of multiple geospatial data sets that are based on a spatial predicate (within, close to, and so on), which can be done through a merge condition.

These spatial functions work in conjunction with the coordinate system specified in Tools > Stream Properties > Options > Geospatial.

Note: These spatial functions do not apply to three-dimensional data. If three-dimensional data is imported into the stream, only the first two dimensions are used by these functions. The z-axis values are ignored.

Table 1. CLEM spatial functions

Function	Result	Description
<code>close_to(SHAPE, SHAPE, NUM)</code>	Boolean	Tests whether 2 shapes are within a certain DISTANCE of each other. If a projected coordinate system is used, DISTANCE is in meters. If no coordinate system is used, it is an arbitrary unit.
<code>crosses(SHAPE, SHAPE)</code>	Boolean	Tests whether 2 shapes cross each other. This function is suitable for 2 linestring shapes, or 1 linestring and 1 polygon.
<code>overlap(SHAPE, SHAPE)</code>	Boolean	Tests whether there is an intersection between 2 polygons and that the intersection is interior to both shapes.
<code>within(SHAPE, SHAPE)</code>	Boolean	Tests whether the entirety of SHAPE1 is contained within a POLYGON.
<code>area(SHAPE)</code>	Real	Returns the area of the specified POLYGON. If a projected system is used, the function returns meters squared. If no coordinate system is used, it is an arbitrary unit. The shape must be a POLYGON or a MULTIPOLYGON.
<code>num_points(SHAPE, LIST)</code>	Integer	Returns the number of points from a point field (MULTIPOINT) which are contained within the bounds of a POLYGON. SHAPE1 must be a POLYGON or a MULTIPOLYGON.
<code>distance(SHAPE, SHAPE)</code>	Real	Returns the distance between SHAPE1 and SHAPE2. If a projected coordinate system is used, the function returns meters. If no coordinate system is used, it is an arbitrary unit. SHAPE1 and SHAPE2 can be any geo measurement type.

Related information

- [Functions reference](#)

Bitwise Integer Operations

These functions enable integers to be manipulated as bit patterns representing two's-complement values, where bit position **N** has weight 2^{**N} . Bits are numbered from 0 upward. These operations act as though the sign bit of an integer is extended indefinitely to the left. Thus, everywhere above its most significant bit, a positive integer has 0 bits and a negative integer has 1 bit.

Table 1. CLEM bitwise integer operations

Function	Result	Description
<code>~~ INT1</code>	Integer	Produces the bitwise complement of the integer INT1. That is, there is a 1 in the result for each bit position for which INT1 has 0. It is always true that <code>~~ INT = -(INT + 1)</code> .
<code>INT1 INT2</code>	Integer	The result of this operation is the bitwise "inclusive or" of INT1 and INT2. That is, there is a 1 in the result for each bit position for which there is a 1 in either INT1 or INT2 or both.
<code>INT1 /& INT2</code>	Integer	The result of this operation is the bitwise "exclusive or" of INT1 and INT2. That is, there is a 1 in the result for each bit position for which there is a 1 in either INT1 or INT2 but not in both.
<code>INT1 && INT2</code>	Integer	Produces the bitwise "and" of the integers INT1 and INT2. That is, there is a 1 in the result for each bit position for which there is a 1 in both INT1 and INT2.
<code>INT1 && ~~ INT2</code>	Integer	Produces the bitwise "and" of INT1 and the bitwise complement of INT2. That is, there is a 1 in the result for each bit position for which there is a 1 in INT1 and a 0 in INT2. This is the same as <code>INT1 && (~~INT2)</code> and is useful for clearing bits of INT1 set in INT2.
<code>INT << N</code>	Integer	Produces the bit pattern of INT1 shifted left by N positions. A negative value for N produces a right shift.
<code>INT >> N</code>	Integer	Produces the bit pattern of INT1 shifted right by N positions. A negative value for N produces a left shift.
<code>INT1 &&= _0 INT2</code>	Boolean	Equivalent to the Boolean expression <code>INT1 && INT2 /== 0</code> but is more efficient.
<code>INT1 &&/_= _0 INT2</code>	Boolean	Equivalent to the Boolean expression <code>INT1 && INT2 == 0</code> but is more efficient.
<code>integer_bitcount(INT)</code>	Integer	Counts the number of 1 or 0 bits in the two's-complement representation of INT. If INT is non-negative, N is the number of 1 bits. If INT is negative, it is the number of 0 bits. Owing to the sign extension, there are an infinite number of 0 bits in a non-negative integer or 1 bits in a negative integer. It is always the case that <code>integer_bitcount(INT) = integer_bitcount(- (INT+1))</code> .

Function	Result	Description
<code>integer_leastbit(INT)</code>	<code>Integer</code>	Returns the bit position N of the least-significant bit set in the integer INT . N is the highest power of 2 by which INT divides exactly.
<code>integer_length(INT)</code>	<code>Integer</code>	Returns the length in bits of INT as a two's-complement integer. That is, N is the smallest integer such that $INT < (1 \ll N) \text{ if } INT \geq 0 \text{ INT } \geq (-1 \ll N) \text{ if } INT < 0$. If INT is non-negative, then the representation of INT as an unsigned integer requires a field of at least N bits. Alternatively, a minimum of $N+1$ bits is required to represent INT as a signed integer, regardless of its sign.
<code>testbit(INT, N)</code>	<code>Boolean</code>	Tests the bit at position N in the integer INT and returns the state of bit N as a Boolean value, which is true for 1 and false for 0.

Related information

- [Functions reference](#)

Random Functions

The following functions are used to randomly select items or randomly generate numbers.

Table 1. CLEM random functions

Function	Result	Description
<code>oneof(LIST)</code>	<code>Any</code>	Returns a randomly chosen element of $LIST$. List items should be entered as <code>[ITEM1, ITEM2, ..., ITEM_N]</code> . Note that a list of field names can also be specified.
<code>random(NUM)</code>	<code>Number</code>	Returns a uniformly distributed random number of the same type (INT or $REAL$), starting from 1 to NUM . If you use an integer, then only integers are returned. If you use a real (decimal) number, then real numbers are returned (decimal precision determined by the stream options). The largest random number returned by the function could equal NUM .
<code>random0(NUM)</code>	<code>Number</code>	This has the same properties as <code>random(NUM)</code> , but starting from 0. The largest random number returned by the function will never equal NUM .

Related information

- [Functions reference](#)

String Functions

In CLEM, you can perform the following operations with strings:

- Compare strings
- Create strings
- Access characters

In CLEM, a string is any sequence of characters between matching double quotation marks ("string quotes"). Characters (**CHAR**) can be any single alphanumeric character. They are declared in CLEM expressions using single backquotes in the form of `<character>`, such as `z`, `A`, or `2`. Characters that are out-of-bounds or negative indices to a string will result in undefined behavior.

Note: Comparisons between strings that do and do not use SQL pushback may generate different results where trailing spaces exist.

Table 1. CLEM string functions

Function	Result	Description
<code>allbutfirst(N, STRING)</code>	<code>String</code>	Returns a string, which is $STRING$ with the first N characters removed.
<code>allbutlast(N, STRING)</code>	<code>String</code>	Returns a string, which is $STRING$ with the last characters removed.
<code>alphabefore(STRING1, STRING2)</code>	<code>Boolean</code>	Used to check the alphabetical ordering of strings. Returns true if $STRING1$ precedes $STRING2$.
<code>endstring(LENGTH, STRING)</code>	<code>String</code>	Extracts the last N characters from the specified string. If the string length is less than or equal to the specified length, then it is unchanged.

Function	Result	Description
<code>hasendstring(STRING, SUBSTRING)</code>	<code>Integer</code>	This function is the same as <code>isendstring(SUBSTRING, STRING)</code> .
<code>hasmidstring(STRING, SUBSTRING)</code>	<code>Integer</code>	This function is the same as <code>ismidstring(SUBSTRING, STRING)</code> (embedded substring).
<code>hasstartswith(STRING, SUBSTRING)</code>	<code>Integer</code>	This function is the same as <code>isstartswith(SUBSTRING, STRING)</code> .
<code>hassubstring(STRING, N, SUBSTRING)</code>	<code>Integer</code>	This function is the same as <code>issubstring(SUBSTRING, N, STRING)</code> , where <code>N</code> defaults to 1.
<code>count_substring(STRING, SUBSTRING)</code>	<code>Integer</code>	Returns the number of times the specified substring occurs within the string. For example, <code>count_substring("foooo.txt", "oo")</code> returns 3.
<code>hassubstring(STRING, SUBSTRING)</code>	<code>Integer</code>	This function is the same as <code>issubstring(SUBSTRING, 1, STRING)</code> , where <code>N</code> defaults to 1.
<code>isalphacode(CHAR)</code>	<code>Boolean</code>	Returns a value of true if <code>CHAR</code> is a character in the specified string (often a field name) whose character code is a letter. Otherwise, this function returns a value of 0. For example, <code>isalphacode(produce_num(1))</code> .
<code>isendstring(SUBSTRING, STRING)</code>	<code>Integer</code>	If the string <code>STRING</code> ends with the substring <code>SUBSTRING</code> , then this function returns the integer subscript of <code>SUBSTRING</code> in <code>STRING</code> . Otherwise, this function returns a value of 0.
<code>islowercode(CHAR)</code>	<code>Boolean</code>	Returns a value of true if <code>CHAR</code> is a lowercase letter character for the specified string (often a field name). Otherwise, this function returns a value of 0. For example, both <code>islowercode(``)</code> and <code>islowercode(country_name(2))</code> are valid expressions.
<code>ismidstring(SUBSTRING, STRING)</code>	<code>Integer</code>	If <code>SUBSTRING</code> is a substring of <code>STRING</code> but does not start on the first character of <code>STRING</code> or end on the last, then this function returns the subscript at which the substring starts. Otherwise, this function returns a value of 0.
<code>isnumbercode(CHAR)</code>	<code>Boolean</code>	Returns a value of true if <code>CHAR</code> for the specified string (often a field name) is a character whose character code is a digit. Otherwise, this function returns a value of 0. For example, <code>isnumbercode(product_id(2))</code> .
<code>isstartswith(SUBSTRING, STRING)</code>	<code>Integer</code>	If the string <code>STRING</code> starts with the substring <code>SUBSTRING</code> , then this function returns the subscript 1. Otherwise, this function returns a value of 0.
<code>issubstring(SUBSTRING, N, STRING)</code>	<code>Integer</code>	Searches the string <code>STRING</code> , starting from its <code>N</code> th character, for a substring equal to the string <code>SUBSTRING</code> . If found, this function returns the integer subscript at which the matching substring begins. Otherwise, this function returns a value of 0. If <code>N</code> is not given, this function defaults to 1.
<code>issubstring(SUBSTRING, STRING)</code>	<code>Integer</code>	Searches the string <code>STRING</code> , starting from its <code>N</code> th character, for a substring equal to the string <code>SUBSTRING</code> . If found, this function returns the integer subscript at which the matching substring begins. Otherwise, this function returns a value of 0. If <code>N</code> is not given, this function defaults to 1.
<code>issubstring_count(SUBSTRING, N, STRING) :</code>	<code>Integer</code>	Returns the index of the <code>N</code> th occurrence of <code>SUBSTRING</code> within the specified <code>STRING</code> . If there are fewer than <code>N</code> occurrences of <code>SUBSTRING</code> , 0 is returned.
<code>issubstring_lim(SUBSTRING, N, STARTLIM, ENDLIM, STRING)</code>	<code>Integer</code>	This function is the same as <code>issubstring</code> , but the match is constrained to start on or before the subscript <code>STARTLIM</code> and to end on or before the subscript <code>ENDLIM</code> . The <code>STARTLIM</code> or <code>ENDLIM</code> constraints may be disabled by supplying a value of false for either argument—for example, <code>issubstring_lim(SUBSTRING, N, false, false, STRING)</code> is the same as <code>issubstring</code> .
<code>isuppercode(CHAR)</code>	<code>Boolean</code>	Returns a value of true if <code>CHAR</code> is an uppercase letter character. Otherwise, this function returns a value of 0. For example, both <code>isuppercode(``)</code> and <code>isuppercode(country_name(2))</code> are valid expressions.
<code>last(CHAR)</code>	<code>String</code>	Returns the last character <code>CHAR</code> of <code>STRING</code> (which must be at least one character long).
<code>length(STRING)</code>	<code>Integer</code>	Returns the length of the string <code>STRING</code> —that is, the number of characters in it.
<code>locchar(CHAR, N, STRING)</code>	<code>Integer</code>	Used to identify the location of characters in symbolic fields. The function searches the string <code>STRING</code> for the character <code>CHAR</code> , starting the search at the <code>N</code> th character of <code>STRING</code> . This function returns a value indicating the location (starting at <code>N</code>) where the character is found. If the character is not found, this function returns a value of 0. If the function has an invalid offset (<code>N</code>) (for example, an offset that is beyond the length of the string), this function returns <code>\$null\$</code> . For example, <code>locchar(`n`, 2, web_page)</code> searches the field called <code>web_page</code> for the <code>`n`</code> character beginning at the second character in the field value. Note: Be sure to use single backquotes to encapsulate the specified character.

Function	Result	Description
<code>locchar_back(CHAR, N, STRING)</code>	<code>Integer</code>	Similar to <code>locchar</code> , except that the search is performed backward starting from the <i>N</i> th character. For example, <code>locchar_back(`n`, 9, web_page)</code> searches the field <code>web_page</code> starting from the ninth character and moving backward toward the start of the string. If the function has an invalid offset (for example, an offset that is beyond the length of the string), this function returns <code>\$null\$</code> . Ideally, you should use <code>locchar_back</code> in conjunction with the function <code>length(<field>)</code> to dynamically use the length of the current value of the field. For example, <code>locchar_back(`n`, (length(web_page)), web_page)</code> .
<code>lowertoupper(CHAR)</code> <code>lowertoupper(STRING)</code>	<code>CHAR or String</code>	Input can be either a string or character, which is used in this function to return a new item of the same type, with any lowercase characters converted to their uppercase equivalents. For example, <code>lowertoupper(`a`)</code> , <code>lowertoupper("My string")</code> , and <code>lowertoupper(field_name(2))</code> are all valid expressions.
<code>matches</code>	<code>Boolean</code>	Returns true if a string matches a specified pattern. The pattern must be a string literal; it cannot be a field name containing a pattern. A question mark (?) can be included in the pattern to match exactly one character; an asterisk (*) matches zero or more characters. To match a literal question mark or asterisk (rather than using these as wildcards), a backslash (\) can be used as an escape character.
<code>replace(SUBSTRING, NEWSUBSTRING, STRING)</code>	<code>String</code>	Within the specified <code>STRING</code> , replace all instances of <code>SUBSTRING</code> with <code>NEWSUBSTRING</code> .
<code>replicate(COUNT, STRING)</code>	<code>String</code>	Returns a string that consists of the original string copied the specified number of times.
<code>stripchar(CHAR, STRING)</code>	<code>String</code>	Enables you to remove specified characters from a string or field. You can use this function, for example, to remove extra symbols, such as currency notations, from data to achieve a simple number or name. For example, using the syntax <code>stripchar(`\$`, 'Cost')</code> returns a new field with the dollar sign removed from all values. Note: Be sure to use single backquotes to encapsulate the specified character.
<code>skipchar(CHAR, N, STRING)</code>	<code>Integer</code>	Searches the string <code>STRING</code> for any character other than <code>CHAR</code> , starting at the <i>N</i> th character. This function returns an integer substring indicating the point at which one is found or 0 if every character from the <i>N</i> th onward is a <code>CHAR</code> . If the function has an invalid offset (for example, an offset that is beyond the length of the string), this function returns <code>\$null\$</code> . <code>locchar</code> is often used in conjunction with the <code>skipchar</code> functions to determine the value of <i>N</i> (the point at which to start searching the string). For example, <code>skipchar(`s`, (locchar(`s`, 1, "MyString")), "MyString")</code> .
<code>skipchar_back(CHAR, N, STRING)</code>	<code>Integer</code>	Similar to <code>skipchar</code> , except that the search is performed backward , starting from the <i>N</i> th character.
<code>startstring(LENGTH, STRING)</code>	<code>String</code>	Extracts the first <i>N</i> characters from the specified string. If the string length is less than or equal to the specified length, then it is unchanged.
<code>strmember(CHAR, STRING)</code>	<code>Integer</code>	Equivalent to <code>locchar(CHAR, 1, STRING)</code> . It returns an integer substring indicating the point at which <code>CHAR</code> first occurs, or 0. If the function has an invalid offset (for example, an offset that is beyond the length of the string), this function returns <code>\$null\$</code> .
<code>subscr(N, STRING)</code>	<code>CHAR</code>	Returns the <i>N</i> th character <code>CHAR</code> of the input string <code>STRING</code> . This function can also be written in a shorthand form as <code>STRING(N)</code> . For example, <code>lowertoupper("name"(1))</code> is a valid expression.
<code>substring(N, LEN, STRING)</code>	<code>String</code>	Returns a string <code>SUBSTRING</code> , which consists of the <i>LEN</i> characters of the string <code>STRING</code> , starting from the character at subscript <i>N</i> .
<code>substring_between(N1, N2, STRING)</code>	<code>String</code>	Returns the substring of <code>STRING</code> , which begins at subscript <i>N1</i> and ends at subscript <i>N2</i> .
<code>trim(STRING)</code>	<code>String</code>	Removes leading and trailing white space characters from the specified string.
<code>trim_start(STRING)</code>	<code>String</code>	Removes leading white space characters from the specified string.
<code>trimend(STRING)</code>	<code>String</code>	Removes trailing white space characters from the specified string.
<code>unicode_char(NUM)</code>	<code>CHAR</code>	Input must be decimal, not hexadecimal values. Returns the character with Unicode value <i>NUM</i> .
<code>unicode_value(CHAR)</code>	<code>NUM</code>	Returns the Unicode value of <code>CHAR</code>
<code>uppertolower(CHAR)</code> <code>uppertolower(STRING)</code>	<code>CHAR or String</code>	Input can be either a string or character and is used in this function to return a new item of the same type with any uppercase characters converted to their lowercase equivalents. Note: Remember to specify strings with double quotes and characters with single backquotes. Simple field names should be specified without quotes.

Related information

- [Functions reference](#)
- [Characters](#)

SoundEx Functions

SoundEx is a method used to find strings when the sound is known but the precise spelling is not. Developed in 1918, it searches out words with similar sounds based on phonetic assumptions about how certain letters are pronounced. It can be used to search names in a database, for example, where spellings and pronunciations for similar names may vary. The basic SoundEx algorithm is documented in a number of sources and, despite known limitations (for example, leading letter combinations such as **ph** and **f** will not match even though they sound the same), is supported in some form by most databases.

Table 1. CLEM soundex functions

Function	Result	Description
soundex (STRING)	<i>Integer</i>	Returns the four-character SoundEx code for the specified <i>STRING</i> .
soundex_difference (STRING1, STRING2)	<i>Integer</i>	Returns an integer between 0 and 4 that indicates the number of characters that are the same in the SoundEx encoding for the two strings, where 0 indicates no similarity and 4 indicates strong similarity or identical strings.

Date and Time Functions

CLEM includes a family of functions for handling fields with datetime storage of string variables representing dates and times. The formats of date and time used are specific to each stream and are specified in the stream properties dialog box. The date and time functions parse date and time strings according to the currently selected format.

When you specify a year in a date that uses only two digits (that is, the century is not specified), IBM® SPSS® Modeler uses the default century that is specified in the stream properties dialog box.

Note: If the data function is pushed back to SQL or IBM SPSS Analytic Server, in a branch that follows an Analytic Server data source, any date format strings (*to_date*) within that data must match the date format specified in the SPSS Modeler stream.

Table 1. CLEM date and time functions

Function	Result	Description
@TODAY	<i>String</i>	If you select Rollover days/mins in the stream properties dialog box, this function returns the current date as a string in the current date format. If you use a two-digit date format and do not select Rollover days/mins, this function returns \$null\$ on the current server.
to_time (ITEM)	<i>Time</i>	Converts the storage of the specified field to a time.
to_date (ITEM)	<i>Date</i>	Converts the storage of the specified field to a date.
to_timestamp (ITEM)	<i>Timestamp</i>	Converts the storage of the specified field to a timestamp.
to_datetime (ITEM)	<i>Datetime</i>	Converts the storage of the specified field to a date, time, or timestamp value.
datetime_date (ITEM)	<i>Date</i>	Returns the date value for a <i>number</i> , <i>string</i> , or <i>timestamp</i> . Note this is the only function that allows you to convert a number (in seconds) back to a date. If <i>ITEM</i> is a string, creates a date by parsing a string in the current date format. The date format specified in the stream properties dialog box must be correct for this function to be successful. If <i>ITEM</i> is a number, it is interpreted as a number of seconds since the base date (or epoch). Fractions of a day are truncated. If <i>ITEM</i> is timestamp, the date part of the timestamp is returned. If <i>ITEM</i> is a date, it is returned unchanged.
date_before (DATE1, DATE2)	<i>Boolean</i>	Returns a value of true if <i>DATE1</i> represents a date or timestamp before that represented by <i>DATE2</i> . Otherwise, this function returns a value of 0.
date_days_difference (DATE1, DATE2)	<i>Integer</i>	Returns the time in days from the date or timestamp represented by <i>DATE1</i> to that represented by <i>DATE2</i> , as an integer. If <i>DATE2</i> is before <i>DATE1</i> , this function returns a negative number.
date_in_days (DATE)	<i>Integer</i>	Returns the time in days from the baseline date to the date or timestamp represented by <i>DATE</i> , as an integer. If <i>DATE</i> is before the baseline date, this function returns a negative number. You must include a valid date for the calculation to work appropriately. For example, you should not specify 29 February 2001 as the date. Because 2001 is not a leap year, this date does not exist.
date_in_months (DATE)	<i>Real</i>	Returns the time in months from the baseline date to the date or timestamp represented by <i>DATE</i> , as a real number. This is an approximate figure based on a month of 30.4375 days. If <i>DATE</i> is before the baseline date, this function returns a negative number. You must include a valid date for the calculation to work appropriately. For example, you should not specify 29 February 2001 as the date. Because 2001 is not a leap year, this date does not exist.

Function	Result	Description
<code>date_in_weeks(DATE)</code>	<code>Real</code>	Returns the time in weeks from the baseline date to the date or timestamp represented by <code>DATE</code> , as a real number. This is based on a week of 7.0 days. If <code>DATE</code> is before the baseline date, this function returns a negative number. You must include a valid date for the calculation to work appropriately. For example, you should not specify 29 February 2001 as the date. Because 2001 is not a leap year, this date does not exist.
<code>date_in_years(DATE)</code>	<code>Real</code>	Returns the time in years from the baseline date to the date or timestamp represented by <code>DATE</code> , as a real number. This is an approximate figure based on a year of 365.25 days. If <code>DATE</code> is before the baseline date, this function returns a negative number. You must include a valid date for the calculation to work appropriately. For example, you should not specify 29 February 2001 as the date. Because 2001 is not a leap year, this date does not exist.
<code>date_months_difference(DATE1, DATE2)</code>	<code>Real</code>	Returns the time in months from the date or timestamp represented by <code>DATE1</code> to that represented by <code>DATE2</code> , as a real number. This is an approximate figure based on a month of 30.4375 days. If <code>DATE2</code> is before <code>DATE1</code> , this function returns a negative number.
<code>datetime_date(YEAR, MONTH, DAY)</code>	<code>Date</code>	Creates a date value for the given <code>YEAR</code> , <code>MONTH</code> , and <code>DAY</code> . The arguments must be integers.
<code>datetime_day(DATE)</code>	<code>Integer</code>	Returns the day of the month from a given <code>DATE</code> or timestamp. The result is an integer in the range 1 to 31.
<code>datetime_day_name(DAY)</code>	<code>String</code>	Returns the full name of the given <code>DAY</code> . The argument must be an integer in the range 1 (Sunday) to 7 (Saturday).
<code>datetime_hour(TIME)</code>	<code>Integer</code>	Returns the hour from a <code>TIME</code> or timestamp. The result is an integer in the range 0 to 23.
<code>datetime_in_seconds(TIME)</code>	<code>Real</code>	Returns the seconds portion stored in <code>TIME</code> .
<code>datetime_in_seconds(DATE), datetime_in_seconds(DATETIME)</code>	<code>Real</code>	Returns the accumulated number, converted into seconds, from the difference between the current <code>DATE</code> or <code>DATETIME</code> and the baseline date (1900-01-01).
<code>datetime_minute(TIME)</code>	<code>Integer</code>	Returns the minute from a <code>TIME</code> or timestamp. The result is an integer in the range 0 to 59.
<code>datetime_month(DATE)</code>	<code>Integer</code>	Returns the month from a <code>DATE</code> or timestamp. The result is an integer in the range 1 to 12.
<code>datetime_month_name(MONTH)</code>	<code>String</code>	Returns the full name of the given <code>MONTH</code> . The argument must be an integer in the range 1 to 12.
<code>datetime_now</code>	<code>Timestamp</code>	Returns the current time as a timestamp.
<code>datetime_second(TIME)</code>	<code>Integer</code>	Returns the second from a <code>TIME</code> or timestamp. The result is an integer in the range 0 to 59.
<code>datetime_day_short_name(DAY)</code>	<code>String</code>	Returns the abbreviated name of the given <code>DAY</code> . The argument must be an integer in the range 1 (Sunday) to 7 (Saturday).
<code>datetime_month_short_name(MONTH)</code>	<code>String</code>	Returns the abbreviated name of the given <code>MONTH</code> . The argument must be an integer in the range 1 to 12.
<code>datetime_time(HOUR, MINUTE, SECOND)</code>	<code>Time</code>	Returns the time value for the specified <code>HOUR</code> , <code>MINUTE</code> , and <code>SECOND</code> . The arguments must be integers.
<code>datetime_time(ITEM)</code>	<code>Time</code>	Returns the time value of the given <code>ITEM</code> .
<code>datetime_timestamp(YEAR, MONTH, DAY, HOUR, MINUTE, SECOND)</code>	<code>Timestamp</code>	Returns the timestamp value for the given <code>YEAR</code> , <code>MONTH</code> , <code>DAY</code> , <code>HOUR</code> , <code>MINUTE</code> , and <code>SECOND</code> .
<code>datetime_timestamp(DATE, TIME)</code>	<code>Timestamp</code>	Returns the timestamp value for the given <code>DATE</code> and <code>TIME</code> .
<code>datetime_timestamp(NUMBER)</code>	<code>Timestamp</code>	Returns the timestamp value of the given number of seconds.
<code>datetime_weeekday(DATE)</code>	<code>Integer</code>	Returns the day of the week from the given <code>DATE</code> or timestamp.
<code>datetime_year(DATE)</code>	<code>Integer</code>	Returns the year from a <code>DATE</code> or timestamp. The result is an integer such as 2002.

Function	Result	Description
<code>date_weeks_difference(DATE1, DATE2)</code>	Real	Returns the time in weeks from the date or timestamp represented by <i>DATE1</i> to that represented by <i>DATE2</i> , as a real number. This is based on a week of 7.0 days. If <i>DATE2</i> is before <i>DATE1</i> , this function returns a negative number.
<code>date_years_difference(DATE1, DATE2)</code>	Real	Returns the time in years from the date or timestamp represented by <i>DATE1</i> to that represented by <i>DATE2</i> , as a real number. This is an approximate figure based on a year of 365.25 days. If <i>DATE2</i> is before <i>DATE1</i> , this function returns a negative number.
<code>date_from_ywd(YEAR, WEEK, DAY)</code>	Integer	Converts the year, week in year, and day in week, to a date using the ISO 8601 standard.
<code>date_iso_day(DATE)</code>	Integer	Returns the day in the week from the date using the ISO 8601 standard.
<code>date_iso_week(DATE)</code>	Integer	Returns the week in the year from the date using the ISO 8601 standard.
<code>date_iso_year(DATE)</code>	Integer	Returns the year from the date using the ISO 8601 standard.
<code>time_before(TIME1, TIME2)</code>	Boolean	Returns a value of true if <i>TIME1</i> represents a time or timestamp before that represented by <i>TIME2</i> . Otherwise, this function returns a value of 0.
<code>time_hours_difference(TIME1, TIME2)</code>	Real	Returns the time difference in hours between the times or timestamps represented by <i>TIME1</i> and <i>TIME2</i> , as a real number. If you select Rollover days/mins in the stream properties dialog box, a higher value of <i>TIME1</i> is taken to refer to the previous day. If you do not select the rollover option, a higher value of <i>TIME1</i> causes the returned value to be negative.
<code>time_in_hours(TIME)</code>	Real	Returns the time in hours represented by <i>TIME</i> , as a real number. For example, under time format <code>HHMM</code> , the expression <code>time_in_hours('0130')</code> evaluates to 1.5. <i>TIME</i> can represent a time or a timestamp.
<code>time_in_mins(TIME)</code>	Real	Returns the time in minutes represented by <i>TIME</i> , as a real number. <i>TIME</i> can represent a time or a timestamp.
<code>time_in_secs(TIME)</code>	Integer	Returns the time in seconds represented by <i>TIME</i> , as an integer. <i>TIME</i> can represent a time or a timestamp.
<code>time_mins_difference(TIME1, TIME2)</code>	Real	Returns the time difference in minutes between the times or timestamps represented by <i>TIME1</i> and <i>TIME2</i> , as a real number. If you select Rollover days/mins in the stream properties dialog box, a higher value of <i>TIME1</i> is taken to refer to the previous day (or the previous hour, if only minutes and seconds are specified in the current format). If you do not select the rollover option, a higher value of <i>TIME1</i> will cause the returned value to be negative.
<code>time_secs_difference(TIME1, TIME2)</code>	Integer	Returns the time difference in seconds between the times or timestamps represented by <i>TIME1</i> and <i>TIME2</i> , as an integer. If you select Rollover days/mins in the stream properties dialog box, a higher value of <i>TIME1</i> is taken to refer to the previous day (or the previous hour, if only minutes and seconds are specified in the current format). If you do not select the rollover option, a higher value of <i>TIME1</i> causes the returned value to be negative.

- [Converting Date and Time Values](#)

Related information

- [Functions reference](#)
- [Conversion Functions](#)
- [Converting Date and Time Values](#)

Converting Date and Time Values

Note that conversion functions (and any other functions that require a specific type of input, such as a date or time value) depend on the current formats specified in the Stream Options dialog box. For example, if you have a field named *DATE* that is stored as a string with values *Jan 2003*, *Feb 2003*, and so on, you could convert it to date storage as follows:

`to_date(DATE)`

For this conversion to work, select the matching date format `MON YYYY` as the default date format for the stream.

For an example that converts string values to dates using a Filler node, see the stream `broadband_create_models.str`, installed in the `|Demos` folder under the `streams` subfolder.

Dates stored as numbers. Note that *DATE* in the previous example is the name of a field, while `to_date` is a CLEM function. If you have dates stored as numbers, you can convert them using the `datetime_date` function, where the number is interpreted as a number of seconds since the base date (or epoch).

`datetime_date(DATE)`

By converting a date to a number of seconds (and back), you can perform calculations such as computing the current date plus or minus a fixed number of days, for example:

```
datetime_date((date_in_days(DATE)-7)*60*60*24)
```

Related information

- [Functions reference](#)
 - [Conversion Functions](#)
 - [Date and Time Functions](#)
-

Sequence functions

For some operations, the sequence of events is important. The application allows you to work with the following record sequences:

- Sequences and time series
- Sequence functions
- Record indexing
- Averaging, summing, and comparing values
- Monitoring change--differentiation
- `@SINCE`
- Offset values
- Additional sequence facilities

For many applications, each record passing through a stream can be considered as an individual case, independent of all others. In such situations, the order of records is usually unimportant.

For some classes of problems, however, the record sequence is very important. These are typically time series situations, in which the sequence of records represents an ordered sequence of events or occurrences. Each record represents a snapshot at a particular instant in time; much of the richest information, however, might be contained not in instantaneous values but in the way in which such values are changing and behaving over time.

Of course, the relevant parameter may be something other than time. For example, the records could represent analyses performed at distances along a line, but the same principles would apply.

Sequence and special functions are immediately recognizable by the following characteristics:

- They are all prefixed by `@`.
- Their names are given in upper case.

Sequence functions can refer to the record currently being processed by a node, the records that have already passed through a node, and even, in one case, records that have yet to pass through a node. Sequence functions can be mixed freely with other components of CLEM expressions, although some have restrictions on what can be used as their arguments.

Examples

You may find it useful to know how long it has been since a certain event occurred or a condition was true. Use the function `@SINCE` to do this—for example:

```
@SINCE(Income > Outgoings)
```

This function returns the offset of the last record where this condition was true—that is, the number of records before this one in which the condition was true. If the condition has never been true, `@SINCE` returns `@INDEX + 1`.

Sometimes you may want to refer to a value of the current record in the expression used by `@SINCE`. You can do this using the function `@THIS`, which specifies that a field name always applies to the current record. To find the offset of the last record that had a `Concentration` field value more than twice that of the current record, you could use:

```
@SINCE(Concentration > 2 * @THIS(Concentration))
```

In some cases the condition given to `@SINCE` is true of the current record by definition—for example:

```
@SINCE(ID == @THIS(ID))
```

For this reason, `@SINCE` does not evaluate its condition for the current record. Use a similar function, `@SINCE0`, if you want to evaluate the condition for the current record as well as previous ones; if the condition is true in the current record, `@SINCE0` returns 0.

Table 1. CLEM sequence functions

Function	Result	Description
----------	--------	-------------

Function	Result	Description
MEAN (FIELD)	<i>Real</i>	Returns the mean average of values for the specified <i>FIELD</i> or <i>FIELDS</i> .
@MEAN (FIELD , EXPR)	<i>Real</i>	Returns the mean average of values for <i>FIELD</i> over the last <i>EXPR</i> records received by the current node, including the current record. <i>FIELD</i> must be the name of a numeric field. <i>EXPR</i> may be any expression evaluating to an integer greater than 0. If <i>EXPR</i> is omitted or if it exceeds the number of records received so far, the average over all of the records received so far is returned.
@MEAN (FIELD , EXPR , INT)	<i>Real</i>	Returns the mean average of values for <i>FIELD</i> over the last <i>EXPR</i> records received by the current node, including the current record. <i>FIELD</i> must be the name of a numeric field. <i>EXPR</i> may be any expression evaluating to an integer greater than 0. If <i>EXPR</i> is omitted or if it exceeds the number of records received so far, the average over all of the records received so far is returned. <i>INT</i> specifies the maximum number of values to look back. This is far more efficient than using just two arguments.
@DIFF1 (FIELD)	<i>Real</i>	Returns the first differential of <i>FIELD</i> . The single-argument form thus simply returns the difference between the current value and the previous value of the field. Returns \$null\$ if the relevant previous records do not exist.
@DIFF1 (FIELD 1 , FIELD2)	<i>Real</i>	The two-argument form gives the first differential of <i>FIELD1</i> with respect to <i>FIELD2</i> . Returns \$null\$ if the relevant previous records do not exist. It is calculated as $\text{@DIFF1}(\text{FIELD1}) / \text{@DIFF1}(\text{FIELD2})$.
@DIFF2 (FIELD)	<i>Real</i>	Returns the second differential of <i>FIELD</i> . The single-argument form thus simply returns the difference between the current value and the previous value of the field. Returns \$null\$ if the relevant previous records do not exist. @DIFF2 is calculated as $\text{@DIFF}(\text{@DIFF}(\text{FIELD}))$.
@DIFF2 (FIELD 1 , FIELD2)	<i>Real</i>	The two-argument form gives the second differential of <i>FIELD1</i> with respect to <i>FIELD2</i> . Returns \$null\$ if the relevant previous records do not exist. This is a complex calculation -- $\text{@DIFF1}(\text{FIELD1}) / \text{@DIFF1}(\text{FIELD2}) - \text{@OFFSET}(\text{@DIFF1}(\text{FIELD1}), 1) / \text{@OFFSET}(\text{@DIFF1}(\text{FIELD2})) / \text{@DIFF1}(\text{FIELD2})$.
@INDEX	<i>Integer</i>	Returns the index of the current record. Indices are allocated to records as they arrive at the current node. The first record is given index 1, and the index is incremented by 1 for each subsequent record.
@LAST_NON_BLANK (FIELD)	<i>Any</i>	Returns the last value for <i>FIELD</i> that was not blank, as defined in an upstream source or Type node. If there are no nonblank values for <i>FIELD</i> in the records read so far, \$null\$ is returned. Note that blank values, also called user-missing values, can be defined separately for each field.
@MAX (FIELD)	<i>Number</i>	Returns the maximum value for the specified <i>FIELD</i> .
@MAX (FIELD , EXPR)	<i>Number</i>	Returns the maximum value for <i>FIELD</i> over the last <i>EXPR</i> records received so far, including the current record. <i>FIELD</i> must be the name of a numeric field. <i>EXPR</i> may be any expression evaluating to an integer greater than 0.
@MAX (FIELD , EXPR , INT)	<i>Number</i>	Returns the maximum value for <i>FIELD</i> over the last <i>EXPR</i> records received so far, including the current record. <i>FIELD</i> must be the name of a numeric field. <i>EXPR</i> may be any expression evaluating to an integer greater than 0. If <i>EXPR</i> is omitted, or if it exceeds the number of records received so far, the maximum value over all of the records received so far is returned. <i>INT</i> specifies the maximum number of values to look back. This is far more efficient than using just two arguments.
@MIN (FIELD)	<i>Number</i>	Returns the minimum value for the specified <i>FIELD</i> .
@MIN (FIELD , EXPR)	<i>Number</i>	Returns the minimum value for <i>FIELD</i> over the last <i>EXPR</i> records received so far, including the current record. <i>FIELD</i> must be the name of a numeric field. <i>EXPR</i> may be any expression evaluating to an integer greater than 0.
@MIN (FIELD , EXPR , INT)	<i>Number</i>	Returns the minimum value for <i>FIELD</i> over the last <i>EXPR</i> records received so far, including the current record. <i>FIELD</i> must be the name of a numeric field. <i>EXPR</i> may be any expression evaluating to an integer greater than 0. If <i>EXPR</i> is omitted, or if it exceeds the number of records received so far, the minimum value over all of the records received so far is returned. <i>INT</i> specifies the maximum number of values to look back. This is far more efficient than using just two arguments.

Function	Result	Description
<code>@OFFSET(FIELD, EXPR)</code>	Any	<p>Returns the value of <i>FIELD</i> in the record offset from the current record by the value of <i>EXPR</i>. A positive offset refers to a record that has already passed (a "lookback"), while a negative one specifies a "lookahead" to a record that has yet to arrive. For example, <code>@OFFSET(Status, 1)</code> returns the value of the <i>Status</i> field in the previous record, while <code>@OFFSET(Status, -4)</code> "looks ahead" four records in the sequence (that is, to records that have not yet passed through this node) to obtain the value. <i>Note that a negative (look ahead) offset must be specified as a constant.</i> For positive offsets only, <i>EXPR</i> may also be an arbitrary CLEM expression, which is evaluated for the current record to give the offset. In this case, the three-argument version of this function should improve performance (see next function). If the expression returns anything other than a non-negative integer, this causes an error—that is, it is not legal to have calculated lookahead offsets. <i>Note: A self-referential @OFFSET function cannot use literal lookahead.</i> For example, in a Filler node, you cannot replace the value of <i>field1</i> using an expression such as <code>@OFFSET(field1, -2)</code>. <i>Note: In the Filler node, when filling a field, there are effectively two different values of that field, namely the pre-filled value and the post-filled value. When @OFFSET refers to itself it refers to the post-filled value. This post-filled value only exists for past rows so self referential @OFFSET can only refer to past rows. Since self referential @OFFSET cannot refer to the future it carries out the following checks of the offset:</i></p> <ul style="list-style-type: none"> • If the offset is literal, and into the future, an error is reported before execution begins. • If the offset is an expression and evaluates to the future at runtime then @OFFSET returns \$null\$. <p><i>Note: Using both "lookahead" and "lookback" within one node is not supported.</i></p>
<code>@OFFSET(FIELD, EXPR, INT)</code>	Any	Performs the same operation as the <code>@OFFSET</code> function with the addition of a third argument, <i>INT</i> , which specifies the maximum number of values to look back. In cases where the offset is computed from an expression, this third argument should improve performance. For example, in an expression such as <code>@OFFSET(Foo, Month, 12)</code> , the system knows to keep only the last twelve values of <i>Foo</i> ; otherwise, it has to store every value just in case. In cases where the offset value is a constant—including negative "lookahead" offsets, which must be constant—the third argument is pointless and the two-argument version of this function should be used. See also the note about self-referential functions in the two-argument version described earlier. <i>Note: Using both "lookahead" and "lookback" within one node is not supported.</i>
<code>@SDEV(FIELD)</code>	Real	Returns the standard deviation of values for the specified <i>FIELD</i> or <i>FIELDS</i> .
<code>@SDEV(FIELD, EXPR)</code>	Real	Returns the standard deviation of values for <i>FIELD</i> over the last <i>EXPR</i> records received by the current node, including the current record. <i>FIELD</i> must be the name of a numeric field. <i>EXPR</i> may be any expression evaluating to an integer greater than 0. If <i>EXPR</i> is omitted, or if it exceeds the number of records received so far, the standard deviation over all of the records received so far is returned.
<code>@SDEV(FIELD, EXPR, INT)</code>	Real	Returns the standard deviation of values for <i>FIELD</i> over the last <i>EXPR</i> records received by the current node, including the current record. <i>FIELD</i> must be the name of a numeric field. <i>EXPR</i> may be any expression evaluating to an integer greater than 0. If <i>EXPR</i> is omitted, or if it exceeds the number of records received so far, the standard deviation over all of the records received so far is returned. <i>INT</i> specifies the maximum number of values to look back. This is far more efficient than using just two arguments.
<code>@SINCE(EXPR)</code>	Any	Returns the number of records that have passed since <i>EXPR</i> , an arbitrary CLEM expression, was true.
<code>@SINCE(EXPR, INT)</code>	Any	Adding the second argument, <i>INT</i> , specifies the maximum number of records to look back. If <i>EXPR</i> has never been true, <i>INT</i> is <code>@INDEX+1</code> .
<code>@SINCE0(EXPR)</code>	Any	Considers the current record, while <code>@SINCE</code> does not; <code>@SINCE0</code> returns 0 if <i>EXPR</i> is true for the current record.
<code>@SINCE0(EXPR, INT)</code>	Any	Adding the second argument, <i>INT</i> specifies the maximum number of records to look back.
<code>@SUM(FIELD)</code>	Number	Returns the sum of values for the specified <i>FIELD</i> or <i>FIELDS</i> .
<code>@SUM(FIELD, EXPR)</code>	Number	Returns the sum of values for <i>FIELD</i> over the last <i>EXPR</i> records received by the current node, including the current record. <i>FIELD</i> must be the name of a numeric field. <i>EXPR</i> may be any expression evaluating to an integer greater than 0. If <i>EXPR</i> is omitted, or if it exceeds the number of records received so far, the sum over all of the records received so far is returned.
<code>@SUM(FIELD, EXPR, INT)</code>	Number	Returns the sum of values for <i>FIELD</i> over the last <i>EXPR</i> records received by the current node, including the current record. <i>FIELD</i> must be the name of a numeric field. <i>EXPR</i> may be any expression evaluating to an integer greater than 0. If <i>EXPR</i> is omitted, or if it exceeds the number of records received so far, the sum over all of the records received so far is returned. <i>INT</i> specifies the maximum number of values to look back. This is far more efficient than using just two arguments.
<code>@THIS(FIELD)</code>	Any	Returns the value of the field named <i>FIELD</i> in the current record. Used only in <code>@SINCE</code> expressions.

Related information

- [Functions reference](#)

Global Functions

The functions @MEAN, @SUM, @MIN, @MAX, and @SDEV work on, at most, all of the records read up to and including the current one. In some cases, however, it is useful to be able to work out how values in the current record compare with values seen in the entire data set. Using a Set Globals node to generate values across the entire data set, you can access these values in a CLEM expression using the global functions.

For example,

`@GLOBAL_MAX(Age)`

returns the highest value of `Age` in the data set, while the expression

`(Value - @GLOBAL_MEAN(Value)) / @GLOBAL_SDEV(Value)`

expresses the difference between this record's `Value` and the global mean as a number of standard deviations. You can use global values only after they have been calculated by a Set Globals node. All current global values can be canceled by clicking the Clear Global Values button on the Globals tab in the stream properties dialog box.

Table 1. CLEM global functions

Function	Result	Description
<code>@GLOBAL_MAX(FIELD)</code>	Number	Returns the maximum value for <i>FIELD</i> over the whole data set, as previously generated by a Set Globals node. <i>FIELD</i> must be the name of a numeric, date/time/datetime, or string field. If the corresponding global value has not been set, an error occurs.
<code>@GLOBAL_MIN(FIELD)</code>	Number	Returns the minimum value for <i>FIELD</i> over the whole data set, as previously generated by a Set Globals node. <i>FIELD</i> must be the name of a numeric, date/time/datetime, or string field. If the corresponding global value has not been set, an error occurs.
<code>@GLOBAL_SDEV(FIELD)</code>	Number	Returns the standard deviation of values for <i>FIELD</i> over the whole data set, as previously generated by a Set Globals node. <i>FIELD</i> must be the name of a numeric field. If the corresponding global value has not been set, an error occurs.
<code>@GLOBAL_MEAN(FIELD)</code>	Number	Returns the mean average of values for <i>FIELD</i> over the whole data set, as previously generated by a Set Globals node. <i>FIELD</i> must be the name of a numeric field. If the corresponding global value has not been set, an error occurs.
<code>@GLOBAL_SUM(FIELD)</code>	Number	Returns the sum of values for <i>FIELD</i> over the whole data set, as previously generated by a Set Globals node. <i>FIELD</i> must be the name of a numeric field. If the corresponding global value has not been set, an error occurs.

Related information

- [Functions reference](#)

Functions Handling Blanks and Null Values

Using CLEM, you can specify that certain values in a field are to be regarded as "blanks," or missing values. The following functions work with blanks.

Table 1. CLEM blank and null value functions

Function	Result	Description
<code>@BLANK(FIELD)</code>	Boolean	Returns true for all records whose values are blank according to the blank-handling rules set in an upstream Type node or source node (Types tab).
<code>@LAST_NON_BLANK(FIELD)</code>	Any	Returns the last value for <i>FIELD</i> that was not blank, as defined in an upstream source or Type node. If there are no nonblank values for <i>FIELD</i> in the records read so far, <code>\$null\$</code> is returned. Note that blank values, also called user-missing values, can be defined separately for each field.
<code>@NULL(FIELD)</code>	Boolean	Returns true if the value of <i>FIELD</i> is the system-missing <code>\$null\$</code> . Returns false for all other values, including user-defined blanks. If you want to check for both, use <code>@BLANK(FIELD)</code> and <code>@NULL(FIELD)</code> .
<code>undef</code>	Any	Used generally in CLEM to enter a <code>\$null\$</code> value—for example, to fill blank values with nulls in the Filler node.

Blank fields may be "filled in" with the Filler node. In both Filler and Derive nodes (multiple mode only), the special CLEM function `@FIELD` refers to the current field(s) being examined.

Related information

- [Functions reference](#)

Special Fields

Special functions are used to denote the specific fields under examination, or to generate a list of fields as input. For example, when deriving multiple fields at once, you should use `@FIELD` to denote "perform this derive action on the selected fields." Using the expression `log(@FIELD)` derives a new log field for each selected field.

Table 1. CLEM special fields

Function	Result	Description
<code>@FIELD</code>	Any	Performs an action on all fields specified in the expression context.
<code>@TARGET</code>	Any	When a CLEM expression is used in a user-defined analysis function, <code>@TARGET</code> represents the target field or "correct value" for the target/predicted pair being analyzed. This function is commonly used in an Analysis node.
<code>@PREDICTED</code>	Any	When a CLEM expression is used in a user-defined analysis function, <code>@PREDICTED</code> represents the predicted value for the target/predicted pair being analyzed. This function is commonly used in an Analysis node.
<code>@PARTITION_FIELD</code>	Any	Substitutes the name of the current partition field.
<code>@TRAINING_PARTITION</code>	Any	Returns the value of the current training partition. For example, to select training records using a Select node, use the CLEM expression: <code>@PARTITION_FIELD = @TRAINING_PARTITION</code> This ensures that the Select node will always work regardless of which values are used to represent each partition in the data.
<code>@TESTING_PARTITION</code>	Any	Returns the value of the current testing partition.
<code>@VALIDATION_PARTITION</code>	Any	Returns the value of the current validation partition.
<code>@FIELDS_BETWEEN(start, end)</code>	Any	Returns the list of field names between the specified start and end fields (inclusive) based on the natural (that is, insert) order of the fields in the data.
<code>@FIELDS_MATCHING(pattern)</code>	Any	Returns a list of field names matching a specified pattern. A question mark (?) can be included in the pattern to match exactly one character; an asterisk (*) matches zero or more characters. To match a literal question mark or asterisk (rather than using these as wildcards), a backslash (\) can be used as an escape character. Note: This requires a string literal as an argument; it cannot use a nested expression to generate the argument.
<code>@MULTI_RESPONSE_SET</code>	Any	Returns the list of fields in the named multiple response set.

Related information

- [Functions reference](#)

Using IBM® SPSS® Modeler with a repository

- [About the IBM SPSS Collaboration and Deployment Services Repository](#)
- [Storing and deploying repository objects](#)
- [Connecting to the Repository](#)
- [Browsing the Repository Contents](#)
- [Storing Objects in the Repository](#)
- [Retrieving Objects from the Repository](#)
- [Searching for objects in the repository](#)
- [Modifying Repository Objects](#)
- [Managing Properties of Repository Objects](#)
- [Deploying streams](#)

About the IBM SPSS Collaboration and Deployment Services Repository

SPSS® Modeler can be used in conjunction with an IBM® SPSS Collaboration and Deployment Services repository, enabling you to manage the life cycle of data mining models and related predictive objects, and enabling these objects to be used by enterprise applications, tools, and solutions.

IBM SPSS Modeler objects that can be shared in this way include streams, nodes, stream outputs, projects, and models. Objects are stored in the central repository, from where they can be shared with other applications and tracked using extended versioning, metadata, and search capabilities.

Before you can use SPSS Modeler with the repository, you need to install an adapter at the repository host. Without this adapter, you may see the following message when attempting to access repository objects from certain SPSS Modeler nodes or models:

The repository may need updating to support new node, model and output types.

For instructions on installing the adapter, see the *SPSS Modeler Deployment Installation* guide, available as a PDF file as part of your product download. Details of how to access IBM SPSS Modeler repository objects from IBM SPSS Deployment Manager are given in the *SPSS Modeler Deployment Guide*.

The following sections provide information on accessing the repository from within SPSS Modeler.

Extensive Versioning and Search Support

The repository provides comprehensive object versioning and search capabilities. For example, suppose that you create a stream and store it in the repository where it can be shared with researchers from other divisions. If you later update the stream in SPSS Modeler, you can add the updated version to the repository without overwriting the previous version. All versions remain accessible and can be searched by name, label, fields used, or other attributes. You could, for example, search for all model versions that use net revenue as an input, or all models created by a particular author. (To do this with a traditional file system, you would have to save each version under a different filename, and the relationships between versions would be unknown to the software.)

Single Sign-On

The single sign-on feature enables users to connect to the repository without having to enter username and password details each time. The user's existing local network login details provide the necessary authentication to IBM SPSS Collaboration and Deployment Services. This feature depends on the following:

- IBM SPSS Collaboration and Deployment Services must be configured to use a single sign-on provider.
- The user must be logged in to a host that is compatible with the provider.

For more information, see [Connecting to the Repository](#).

Related information

- [Connecting to the Repository](#)
- [Entering Credentials for the Repository](#)
- [Browsing the Repository Contents](#)
- [Storing Objects in the Repository](#)
- [Retrieving Objects from the Repository](#)
- [Searching for objects in the repository](#)
- [Creating, Renaming, and Deleting Folders](#)
- [Viewing Folder Properties](#)
- [Viewing and Editing Object Properties](#)

Storing and deploying repository objects

Streams created in IBM® SPSS® Modeler can be **stored** in the repository just as they are, as files with the extension .str. In this way, a single stream can be accessed by multiple users throughout the enterprise. See the topic [Storing Objects in the Repository](#) for more information.

It is also possible to deploy a stream in the repository. A deployed stream is stored as a file with additional metadata. A deployed stream can take full advantage of the enterprise-level features of IBM SPSS Collaboration and Deployment Services, such as automated scoring and model refresh. For example, a model can be automatically updated at regularly-scheduled intervals as new data becomes available. Alternatively, a set of streams can be deployed for Champion Challenger analysis, in which streams are compared to determine which one contains the most effective predictive model.

Note: The Association Rules, STP, and TCM modeling nodes do not support the Model Evaluation or Champion Challenger steps in IBM SPSS Collaboration and Deployment Services.

You can deploy a stream (with the extension .str). Deployment as a stream enables the stream to be used by the thin-client application IBM SPSS Modeler Advantage. See the topic [Opening a Stream in IBM SPSS Modeler Advantage](#) for more information.

For more information, see [Stream Deployment Options](#).

Other Deployment Options

While IBM SPSS Collaboration and Deployment Services offers the most extensive features for managing enterprise content, a number of other mechanisms for deploying or exporting streams are also available, including:

- Export the stream and model for later use with IBM SPSS Modeler Solution Publisher Runtime.
- Export one or more models in PMML, an XML-based format for encoding model information. See the topic [Importing and exporting models as PMML](#) for more information.

Related information

- [About IBM SPSS Modeler](#)
- [IBM SPSS Modeler Server](#)
- [Documentation](#)
- [Types of models](#)

Connecting to the Repository

1. To connect to the repository, on the IBM® SPSS® Modeler main menu, click:
Tools > Repository > Options...
2. In the RepositoryURL. field, enter or select the directory path to, or URL of, the repository installation you want to access. You can connect to only one repository at a time.
Settings are specific to each site or installation. For specific login details, contact your local system administrator.

Set Credentials. Leave this box unchecked to enable the *single sign-on* feature, which attempts to log you in using your local computer username and password details. If single sign-on is not possible, or if you select this option to disable single sign-on (for example, to log in to an administrator account), a screen is displayed for you to enter your credentials.

- [Entering Credentials for the Repository](#)
- [Browse for repository credentials](#)

Related information

- [About the IBM SPSS Collaboration and Deployment Services Repository](#)
- [Entering Credentials for the Repository](#)
- [Browsing the Repository Contents](#)
- [Storing Objects in the Repository](#)
- [Retrieving Objects from the Repository](#)
- [Searching for objects in the repository](#)
- [Creating, Renaming, and Deleting Folders](#)
- [Viewing Folder Properties](#)
- [Viewing and Editing Object Properties](#)

Entering Credentials for the Repository

Depending on your settings, the following fields may be required in the Repository: Credentials dialog box:

User ID and password. Specify a valid user name and password for logging on. If necessary, contact your local administrator for more information.

Provider. Choose a security provider for authentication. The repository can be configured to use different security providers; if necessary, contact your local administrator for more information.

Remember repository and user ID. Saves the current settings as the default so that you do not have to reenter them each time you want to connect.

Related information

- [About the IBM SPSS Collaboration and Deployment Services Repository](#)
- [Connecting to the Repository](#)
- [Browsing the Repository Contents](#)

- [Storing Objects in the Repository](#)
 - [Retrieving Objects from the Repository](#)
 - [Searching for objects in the repository](#)
 - [Creating, Renaming, and Deleting Folders](#)
 - [Viewing Folder Properties](#)
 - [Viewing and Editing Object Properties](#)
-

Browse for repository credentials

When you connect to a repository from an Analytic Server, Cognos, ODBC, or TM1 source node, you can select previously recorded credentials to connect to a repository. These credentials are listed in the Select Repository Credential dialog box. To select this dialog box, click Browse next to the Credential field.

In the Select Repository Credential dialog box, highlight the credentials in the list supplied and click OK. If the list is too large, use the Filter field to enter the name, or part of the name, to find the credentials you require.

Related information

- [About the IBM SPSS Collaboration and Deployment Services Repository](#)
 - [Connecting to the Repository](#)
 - [Browsing the Repository Contents](#)
 - [Storing Objects in the Repository](#)
 - [Retrieving Objects from the Repository](#)
 - [Searching for objects in the repository](#)
 - [Creating, Renaming, and Deleting Folders](#)
 - [Viewing Folder Properties](#)
 - [Viewing and Editing Object Properties](#)
-

Browsing the Repository Contents

The repository allows you to browse stored content in a manner similar to Windows Explorer; you can also browse *versions* of each stored object.

1. To open the IBM® SPSS® Collaboration and Deployment Services Repository window, on the SPSS Modeler menus click:

Tools > Repository > Explore...

1. Specify connection settings to the repository if necessary. See the topic [Connecting to the Repository](#) for more information. For specific port, password, and other connection details, contact your local system administrator.

The explorer window initially displays a tree view of the folder hierarchy. Click a folder name to display its contents.

Objects that match the current selection or search criteria are listed in the right pane, with detailed information on the selected version displayed in the lower right pane. The attributes displayed apply to the most recent version.

Related information

- [About the IBM SPSS Collaboration and Deployment Services Repository](#)
 - [Connecting to the Repository](#)
 - [Entering Credentials for the Repository](#)
 - [Storing Objects in the Repository](#)
 - [Retrieving Objects from the Repository](#)
 - [Searching for objects in the repository](#)
 - [Creating, Renaming, and Deleting Folders](#)
 - [Viewing Folder Properties](#)
 - [Viewing and Editing Object Properties](#)
-

Storing Objects in the Repository

You can store streams, nodes, models, model palettes, projects, and output objects in the repository, from where they can be accessed by other users and applications.

You can also publish stream output to the repository in a format that enables other users to view it over the Internet using the IBM® SPSS® Collaboration and Deployment Services Deployment Portal.

- [Setting Object Properties](#)
- [Storing Streams](#)
- [Storing Projects](#)
- [Storing Nodes](#)
- [Storing Output Objects](#)
- [Storing Models and Model Palettes](#)

Related information

- [About the IBM SPSS Collaboration and Deployment Services Repository](#)
 - [Connecting to the Repository](#)
 - [Entering Credentials for the Repository](#)
 - [Browsing the Repository Contents](#)
 - [Retrieving Objects from the Repository](#)
 - [Searching for objects in the repository](#)
 - [Creating, Renaming, and Deleting Folders](#)
 - [Viewing Folder Properties](#)
 - [Viewing and Editing Object Properties](#)
-

Setting Object Properties

When you store an object, the Repository: Store dialog box is displayed, enabling you to set the values of a number of properties for the object. You can:

- Choose the name and repository folder under which the object is to be stored
- Add information about the object such as the version label and other searchable properties
- Assign one or more classification topics to the object
- Set security options for the object

The following sections describe the properties you can set.

- [Choosing the Location for Storing Objects](#)
 - [Adding Information About Stored Objects](#)
 - [Assigning Topics to a Stored Object](#)
 - [Setting Security Options for Stored Objects](#)
-

Choosing the Location for Storing Objects

In the Repository: Store dialog box, enter the following.

Save in. Shows the current folder--the location where the object will be stored. Double-click a folder name in the list to set that folder as the current folder. Use the Up Folder button to navigate to the parent folder. Use the New Folder button to create a folder at the current level.

File name. The name under which the object will be stored.

Store. Stores the object at the current location.

Adding Information About Stored Objects

All of the fields on the Information tab of the Repository: Store dialog box are optional.

Author. The username of the user creating the object in the repository. By default, this shows the username used for the repository connection, but you can change this name here.

Version Label. Select a label from the list to indicate the object version, or click Add to create a new label. Avoid using the "[" character in the label. Ensure that no boxes are checked if you do not want to assign a label to this object version. See the topic [Viewing and Editing Object Properties](#) for more information.

Description. A description of the object. Users can search for objects by description (see note).

Keywords. One or more keywords that relate to the object and which can be used for search purposes (see note).

Expiration. A date after which the object is no longer visible to general users, although it can still be seen by its owner and by the repository administrator. To set an expiration date, select the Date option and enter the date, or choose one using the calendar button.

Store. Stores the object at the current location.

Note: Information in the Description and Keywords fields is treated as distinct from anything entered in SPSS® Modeler on the Annotations tab of the object. A repository search by description or keyword does not return information from the Annotations tab. See the topic [Searching for objects in the repository](#) for more information.

Assigning Topics to a Stored Object

Topics are a hierarchical classification system for the content stored in the repository. You can choose from the available topics when storing objects, and users can also search for objects by topic. The list of available topics is set by repository users with the appropriate privileges (for more information, see the *Deployment Manager User's Guide*).

To assign a topic to the object, on the Topics tab of the Repository: Store dialog box:

1. Click the Add button.
2. Click a topic name from the list of available topics.
3. Click OK.

To remove a topic assignment:

4. Select the topic in the list of assigned topics.
5. Click Delete.

Setting Security Options for Stored Objects

You can set or change a number of security options for a stored object on the Security tab of the Repository: Store dialog box. For one or more **principals** (that is, users or groups of users), you can:

- Assign access rights to the object
- Modify access rights to the object
- Remove access rights to the object

Principal. The repository username of the user or group who has access rights on this object.

Permissions. The access rights that this user or group has for the object.

Add. Enables you to add one or more users or groups to the list of those with access rights on this object. See the topic [Adding a User to the Permissions List](#) for more information.

Modify. Enables you to modify the access rights of the selected user or group for this object. Read access is granted by default. This option enables you to grant additional access rights, namely Owner, Write, Delete, and Modify Permissions.

Delete. Deletes the selected user or group from the permissions list for this object.

- [Adding a User to the Permissions List](#)
- [Modifying Access Rights for an Object](#)

Adding a User to the Permissions List

The following fields are available when you select Add on the Security tab of the Repository: Store dialog box:

Select provider. Choose a security provider for authentication. The repository can be configured to use different security providers; if necessary, contact your local administrator for more information.

Find. Enter the repository username of the user or group you want to add, and click Search to display that name in the user list. To add more than one username at a time, leave this field blank and just click Search to display a list of all the repository usernames.

User list. Select one or more usernames from the list and click OK to add them to the permissions list.

Modifying Access Rights for an Object

The following fields are available when you select Modify on the Security tab of the Repository: Store dialog box:

Owner. Select this option to give this user or group owner access rights to the object. The owner has full control over the object, including Delete and Modify access rights.

Read. By default, a user or group that is not the object owner has only Read access rights to the object. Select the appropriate check boxes to add Write, Delete, and Modify Permissions access rights for this user or group.

Storing Streams

You can store a stream as a .str file in the repository, from where it can be accessed by other users.

Note: For information on deploying a stream, to take advantage of additional repository features, see [Deploying streams](#).

To store the current stream:

1. On the main menu, click:
`File > Store > Store as Stream...`
2. Specify connection settings to the repository if necessary. See the topic [Connecting to the Repository](#) for more information. For specific port, password, and other connection details, contact your local system administrator.
3. In the Repository: Store dialog box, choose the folder where you want to store the object, specify any other information you want to record, and click the Store button. See the topic [Setting Object Properties](#) for more information.

Storing Projects

You can store a complete IBM® SPSS® Modeler project as a .cpj file in the repository so that it can be accessed by other users.

Because a project file is a container for other IBM SPSS Modeler objects, you need to tell IBM SPSS Modeler to store the project's objects in the repository. You do this using a setting in the Project Properties dialog box. See the topic [Setting Project Properties](#) for more information.

Once you configure a project to store objects in the repository, whenever you add a new object to the project, IBM SPSS Modeler automatically prompts you to store the object.

When you have finished your IBM SPSS Modeler session, you must store a new version of the project file so that it remembers your additions. The project file automatically contains (and retrieves) the latest versions of its objects. If you did not add any objects to a project during an IBM SPSS Modeler session, then you do not have to re-store the project file. You must, however, store new versions for the project objects (streams, output, and so forth) that you changed.

To store a project

1. Select the project on the CRISP-DM or Classes tab in the managers pane in IBM SPSS Modeler, and on the main menu click:
`File > Project > Store Project...`
2. Specify connection settings to the repository if necessary. See the topic [Connecting to the Repository](#) for more information. For specific port, password, and other connection details, contact your local system administrator.
3. In the Repository: Store dialog box, choose the folder where you want to store the object, specify any other information you want to record, and click the Store button. See the topic [Setting Object Properties](#) for more information.

Storing Nodes

You can store an individual node definition from the current stream as a .nod file in the repository, from where it can be accessed by other users.

To store a node:

1. Right-click the node in the stream canvas and click Store Node.
2. Specify connection settings to the repository if necessary. See the topic [Connecting to the Repository](#) for more information. For specific port, password, and other connection details, contact your local system administrator.
3. In the Repository: Store dialog box, choose the folder where you want to store the object, specify any other information you want to record, and click the Store button. See the topic [Setting Object Properties](#) for more information.

Storing Output Objects

You can store an output object from the current stream as a .cou file in the repository, from where it can be accessed by other users.

To store an output object:

1. Click the object on the Outputs tab of the managers pane in SPSS® Modeler, and on the main menu click:
File->Outputs->Store Output...
2. Alternatively, right-click an object in the Outputs tab and click Store.
3. Specify connection settings to the repository if necessary. See the topic [Connecting to the Repository](#) for more information. For specific port, password, and other connection details, contact your local system administrator.
4. In the Repository: Store dialog box, choose the folder where you want to store the object, specify any other information you want to record, and click the Store button. See the topic [Setting Object Properties](#) for more information.

Storing Models and Model Palettes

You can store an individual model as a .gm file in the repository, from where it can be accessed by other users. You can also store the complete contents of the Models palette as a .gen file in the repository.

Storing a model:

1. Click the object on the Models palette in SPSS® Modeler, and on the main menu click:
File->Models->Store Model...
2. Alternatively, right-click an object in the Models palette and click Store Model.
3. Continue from "Completing the storage procedure".

Storing a Models palette:

1. Right-click the background of the Models palette.
2. On the pop-up menu, click Store Palette.
3. Continue from "Completing the storage procedure".

Completing the storage procedure:

1. Specify connection settings to the repository if necessary. See the topic [Connecting to the Repository](#) for more information. For specific port, password, and other connection details, contact your local system administrator.
2. In the Repository: Store dialog box, choose the folder where you want to store the object, specify any other information you want to record, and click the Store button. See the topic [Setting Object Properties](#) for more information.

Retrieving Objects from the Repository

You can retrieve streams, models, model palettes, nodes, projects, and output objects that have been stored in the repository.

Note: Besides using the menu options as described here, you can also retrieve streams, output objects, models and model palettes by right-clicking in the appropriate tab of the managers pane at the top right of the SPSS® Modeler window.

1. To retrieve a stream, on the IBM® SPSS Modeler main menu click:
File->Retrieve Stream...
2. To retrieve a model, model palette, project, or output object, on the IBM SPSS Modeler main menu click:
File->Models->Retrieve Model...
or
File->Models->Retrieve Models Palette...
or
File->Projects->Retrieve Project...
or
File->Outputs->Retrieve Output...

3. Alternatively, right-click in the managers or project pane and click Retrieve on the pop-up menu.
4. To retrieve a node, on the IBM SPSS Modeler main menu click:
Insert > Node (or SuperNode) from Repository...
 - a. Specify connection settings to the repository if necessary. See the topic [Connecting to the Repository](#) for more information. For specific port, password, and other connection details, contact your local system administrator.
5. In the Repository: Retrieve dialog box, browse to the object, select it and click the Retrieve button. See the topic [Choosing an Object to Retrieve](#) for more information.
 - [Choosing an Object to Retrieve](#)
 - [Selecting an Object Version](#)

Related information

- [About the IBM SPSS Collaboration and Deployment Services Repository](#)
 - [Connecting to the Repository](#)
 - [Entering Credentials for the Repository](#)
 - [Browsing the Repository Contents](#)
 - [Storing Objects in the Repository](#)
 - [Searching for objects in the repository](#)
 - [Creating, Renaming, and Deleting Folders](#)
 - [Viewing Folder Properties](#)
 - [Viewing and Editing Object Properties](#)
-

Choosing an Object to Retrieve

The following fields are available in the Repository: Retrieve/Search dialog box:

Look in. Shows the folder hierarchy for the current folder. To navigate to a different folder, select one from this list to navigate there directly, or navigate using the object list below this field.

Up Folder button. Navigates to one level above the current folder in the hierarchy.

New Folder button. Creates a new folder at the current level in the hierarchy.

File name. The repository file name of the selected object. To retrieve that object, click Retrieve.

Files of type. The type of object that you have chosen to retrieve. Only objects of this type, together with folders, are shown in the object list. To display objects of a different type for retrieval, select the object type from the list.

Open as locked. By default, when an object is retrieved, it is locked in the repository so that others cannot update it. If you do not want the object to be locked on retrieval, uncheck this box.

Description, Keywords. If additional details about the object were defined when the object was stored, those details are displayed here. See the topic [Adding Information About Stored Objects](#) for more information.

Version. To retrieve a version of the object other than the latest, click this button. Detailed information for all versions is displayed, allowing you to choose the version you want.

Selecting an Object Version

To select a specific version of a repository object, in the Repository: Select Version dialog box:

1. (Optional) Sort the list by version, label, size, creation date or creating user, by double-clicking on the header of the appropriate column.
 2. Select the object version you want to work with.
 3. Click Continue.
-

Searching for objects in the repository

You can search for objects by name, folder, type, label, date, or other criteria.

Searching for objects by name

-
1. On the IBM® SPSS® Modeler main menu click:
Tools > Repository > Explore...
 - a. Specify connection settings to the repository if necessary. See the topic [Connecting to the Repository](#) for more information. For specific port, password, and other connection details, contact your local system administrator.
 2. Click the Search tab.
 3. In the Search for objects named field, specify the name of the object you want to find.

When searching for objects by name, an asterisk (*) can be used as a wildcard character to match any string of characters, and a question mark (?) matches any single character. For example, *cluster* matches all objects that include the string cluster anywhere in the name. The search string m0?_* matches M01_cluster.str and M02_cluster.str but not M01a_cluster.str. Searches are not case sensitive (cluster matches Cluster matches CLUSTER).

Note: If the number of objects is large, searches may take a few moments.

Searching by other criteria

You can perform a search based on title, label, dates, author, keywords, indexed content, or description. Only objects that match *all* specified search criteria will be found. For example, you could locate all streams containing one or more clustering models that also have a specific label applied, and which were modified after a specific date.

Object Types. You can restrict the search to models, streams, outputs, nodes, SuperNodes, projects, model palettes, or other types of objects.

- **Models.** You can search for models by category (classification, approximation, clustering, etc.) or by a specific modeling algorithm, such as Kohonen.
You can also search by fields used—for example, all models that use a field named *income* as an input or output (target) field.
- **Streams.** For streams, you can restrict the search by fields used, or model type (either category or algorithm) contained in the stream.

Topics. You can search on models associated with specific topics from a list set by repository users with the appropriate privileges (for more information, see the *Deployment Manager User's Guide*). To obtain the list, check this box, then click the Add Topics button that is displayed, select one or more topics from the list and click OK.

Label. Restricts the search to specific object version labels.

Dates. You can specify a creation or modification date and search on objects before, after, or between the specified date range.

Author. Restricts the search to objects created by a specific user.

Keywords. Search on specific keywords. In IBM SPSS Modeler, keywords are specified on the Annotation tab for a stream, model, or output object.

Description. Search on specific terms in the description field. In IBM SPSS Modeler, the description is specified on the Annotation tab for a stream, model, or output object. Multiple search phrases can be separated by semicolons—for example, *income*; *crop type*; *claim value*. (Note that within a search phrase, spaces matter. For example, *crop type* with one space and *crop type* with two spaces are not the same.)

Related information

- [About the IBM SPSS Collaboration and Deployment Services Repository](#)
- [Connecting to the Repository](#)
- [Entering Credentials for the Repository](#)
- [Browsing the Repository Contents](#)
- [Storing Objects in the Repository](#)
- [Retrieving Objects from the Repository](#)
- [Creating, Renaming, and Deleting Folders](#)
- [Viewing Folder Properties](#)
- [Viewing and Editing Object Properties](#)

Modifying Repository Objects

You can modify existing objects in the repository directly from SPSS® Modeler. You can:

- Create, rename, or delete folders
 - Lock or unlock objects
 - Delete objects
- [Creating, Renaming, and Deleting Folders](#)

- [Locking and Unlocking Repository Objects](#)
 - [Deleting Repository Objects](#)
-

Creating, Renaming, and Deleting Folders

1. To perform operations on folders in the repository, on the SPSS® Modeler main menu click:
Tools > Repository > Explore...
 - a. Specify connection settings to the repository if necessary. See the topic [Connecting to the Repository](#) for more information. For specific port, password, and other connection details, contact your local system administrator.
2. Ensure that the Folders tab is active.
3. To create a new folder, right-click the parent folder and click New Folder.
4. To rename a folder, right-click it and click Rename Folder.
5. To delete a folder, right-click it and click Delete Folder.

Related information

- [About the IBM SPSS Collaboration and Deployment Services Repository](#)
 - [Connecting to the Repository](#)
 - [Entering Credentials for the Repository](#)
 - [Browsing the Repository Contents](#)
 - [Storing Objects in the Repository](#)
 - [Retrieving Objects from the Repository](#)
 - [Searching for objects in the repository](#)
 - [Viewing Folder Properties](#)
 - [Viewing and Editing Object Properties](#)
-

Locking and Unlocking Repository Objects

You can lock an object to prevent other users from updating any of its existing versions or creating new versions. A locked object is indicated by a padlock symbol over the object icon.

Figure 1. Locked object



To lock an object

1. In the repository explorer window, right-click the required object.
2. Click Lock.

To unlock an object

1. In the repository explorer window, right-click the required object.
2. Click Unlock.

Deleting Repository Objects

Before deleting an object from the repository, you must decide if you want to delete all versions of the object, or just a particular version.

To Delete All Versions of an Object

1. In the repository explorer window, right-click the required object.
2. Click Delete Objects.

To Delete the Most Recent Version of an Object

1. In the repository explorer window, right-click the required object.
2. Click Delete.

To Delete a Previous Version of an Object

1. In the repository explorer window, right-click the required object.
2. Click Delete Versions.

3. Select the version(s) to delete and click OK.

Managing Properties of Repository Objects

You can control various object properties from SPSS® Modeler. You can:

- View the properties of a folder
- View and edit the properties of an object
- Create, apply and delete version labels for an object
- [Viewing Folder Properties](#)
- [Viewing and Editing Object Properties](#)
- [Managing Object Version Labels](#)

Viewing Folder Properties

To view properties for any folder in the repository window, right-click the required folder. Click Folder Properties.

General tab

This tab displays the folder name, creation, and modification dates.

Permissions tab

In this tab you specify read and write permissions for the folder. All users and groups with access to the parent folder are listed. Permissions follow a hierarchy. For example, if you do not have read permission, you cannot have write permission. If you do not have write permission, you cannot have delete permission.

Users And Groups. Lists the repository users and groups that have at least Read access to this folder. Select the Write and Delete check boxes to add those access rights for this folder to a particular user or group. Click the Add Users/Groups icon on the right side of the Permissions tab to assign access to additional users and groups. The list of available users and groups is controlled by the administrator.

Cascade Permissions. Choose an option to control how changes made to the current folder are applied to its child folders, if any.

- **Cascade all permissions.** Cascades permission settings from the current folder to all child and descendant folders. This is a quick way to set permissions for several folders at once. Set permissions as required for the parent folder, and then cascade as required.
- **Cascade changes only.** Cascades only changes made since the last time changes were applied. For example, if a new group has been added and you want to give it access to all folders under the Sales branch, you can give the group access to the root Sales folder and cascade the change to all subfolders. All other permissions to existing subfolders remain as before.
- **Do not cascade.** Any changes made apply to the current folder only and do not cascade to child folders.

Related information

- [About the IBM SPSS Collaboration and Deployment Services Repository](#)
- [Connecting to the Repository](#)
- [Entering Credentials for the Repository](#)
- [Browsing the Repository Contents](#)
- [Storing Objects in the Repository](#)
- [Retrieving Objects from the Repository](#)
- [Searching for objects in the repository](#)
- [Creating, Renaming, and Deleting Folders](#)
- [Viewing and Editing Object Properties](#)

Viewing and Editing Object Properties

In the Object Properties dialog box you can view and edit properties. Although some properties cannot be changed, you can always update an object by adding a new version.

1. In the repository window, right-click the required object.
2. Click Object Properties.

General Tab

Name. The name of the object as viewed in the repository.

Created on. Date the object (not the version) was created.

Last modified. Date the most recent version was modified.

Author. The user's login name.

Description. By default, this contains the description specified on the object's Annotation tab in SPSS® Modeler.

Linked topics. The repository allows models and related objects to be organized by topics if required. The list of available topics is set by repository users with the appropriate privileges (for more information, see the *Deployment Manager User's Guide*).

Keywords. You specify keywords on the Annotation tab for a stream, model, or output object. Multiple keywords should be separated by spaces, up to a maximum of 255 characters. (If keywords contain spaces, use quotation marks to separate them.)

Versions Tab

Objects stored in the repository may have multiple versions. The Versions tab displays information about each version.

The following properties can be specified or modified for specific versions of a stored object:

Version. Unique identifier for the version generated based on the time when the version was stored.

Label. Current label for the version, if any. Unlike the version identifier, labels can be moved from one version of an object to another.

The file size, creation date, and author are also displayed for each version.

Edit Labels. Click the Edit Labels icon at the top right of the Versions tab to define, apply or remove labels for stored objects. See the topic [Managing Object Version Labels](#) for more information.

Permissions Tab

On the Permissions tab you can set read and write permissions for the object. All users and groups with access to the current object are listed. Permissions follow a hierarchy. For example, if you do not have read permission, you cannot have write permission. If you do not have write permission, you cannot have delete permission.

Users And Groups. Lists the repository users and groups that have at least Read access to this object. Select the Write and Delete check boxes to add those access rights for this object to a particular user or group. Click the Add Users/Groups icon on the right side of the Permissions tab to assign access to additional users and groups. The list of available users and groups is controlled by the administrator.

Related information

- [About the IBM SPSS Collaboration and Deployment Services Repository](#)
 - [Connecting to the Repository](#)
 - [Entering Credentials for the Repository](#)
 - [Browsing the Repository Contents](#)
 - [Storing Objects in the Repository](#)
 - [Retrieving Objects from the Repository](#)
 - [Searching for objects in the repository](#)
 - [Creating, Renaming, and Deleting Folders](#)
 - [Viewing Folder Properties](#)
-

Managing Object Version Labels

The Edit Version Labels dialog box enables you to:

- Apply labels to the selected object
- Remove labels from the selected object
- Define a new label and apply it to the object

To apply labels to the object

1. Select one or more labels in the Available Labels list.
2. Click the right-arrow button to move the selected labels to the Applied Labels list.
3. Click OK.

To remove labels from the object

1. Select one or more labels in the Applied Labels list.

2. Click the left-arrow button to move the selected labels to the Available Labels list.
3. Click OK.

To define a new label and apply it to the object

1. Type the label name in the New Label field.
2. Click the right-arrow button to move the new label to the Applied Labels list.
3. Click OK.

Deploying streams

To enable a stream to be used with the thin-client application IBM® SPSS® Modeler Advantage, it must be deployed as a stream (.str file) in the repository.

Note: You cannot deploy a stream that has more than one source node in the scoring branch.

The stream can take full advantage of the enterprise-level features of IBM SPSS Collaboration and Deployment Services. See the topic [Storing and deploying repository objects](#) for more information.

To deploy the current stream (File menu method)

1. On the main menu, click:
File > Store > Deploy
2. Choose the deployment type and complete the rest of the dialog box as necessary.
3. Click Deploy as stream to deploy the stream for use with IBM SPSS Modeler Advantage or IBM SPSS Collaboration and Deployment Services.
4. Click Store. For more information, click Help.
5. Continue from "Completing the deployment process."

To deploy the current stream (Tools menu method)

1. On the main menu, click:
Tools > Stream Properties > Deployment
2. Choose the deployment type, complete the rest of the Deployment tab as necessary, and click Store. See the topic [Stream Deployment Options](#) for more information.

Completing the deployment process

1. Specify connection settings to the repository if necessary. See the topic [Connecting to the Repository](#) for more information. For specific port, password, and other connection details, contact your local system administrator.
2. In the Repository: Store dialog box, choose the folder where you want to store the object, specify any other information you want to record, and click Store. See the topic [Setting Object Properties](#) for more information.
 - [Stream Deployment dialog](#)
 - [Stream Deployment Options](#)
 - [The Scoring Branch](#)

Stream Deployment dialog

On this dialog you specify whether you want to deploy the stream for scoring purposes only, or for model refresh as well. If you choose scoring only, you just need to designate a terminal node as the scoring node. For model refresh, you need to designate additional nodes to enable the model to be refreshed.

You also specify on this screen whether you want to deploy as a stream (for use with IBM® SPSS® Modeler Advantage or IBM SPSS Collaboration and Deployment Services).

Deployment type. Choose how you want to deploy the stream. All streams require a designated scoring node before they can be deployed; additional requirements and options depend on the deployment type.

Note: The Association Rules, STP, and TCM modeling nodes do not support the Model Evaluation or Champion Challenger steps in IBM SPSS Collaboration and Deployment Services.

- <none>. The stream will not be deployed to the repository. All options are disabled except stream description preview.

- Scoring Only. The stream is deployed to the repository when you click the Store button. Data can be scored using the node that you designate in the Scoring node field.
- Model Refresh. Same as for Scoring Only but in addition, the model can be updated in the repository using the objects that you designate in the Modeling node and Model nugget fields. Note that automatic model refresh is not supported by default in IBM SPSS Collaboration and Deployment Services, so you must choose this deployment type if you want to use this feature when running a stream from the repository. See [Model Refresh](#) for more information.

Scoring node. Select a graph, output or export node to identify the stream branch to be used for scoring the data. While the stream can actually contain any number of valid branches, models, and terminal nodes, one and only one scoring branch must be designated for purposes of deployment. This is the most basic requirement to deploy any stream.

Scoring Parameters. Allows you to specify parameters that can be modified when the scoring branch is run. See [Scoring and modeling parameters](#) for more information.

Modeling node. For model refresh, specifies the modeling node used to regenerate or update the model in the repository. Must be a modeling node of the same type as that specified for Model nugget.

Model Build Parameters. Allows you to specify parameters that can be modified when the modeling node is run. See [Scoring and modeling parameters](#) for more information.

Model nugget. For model refresh, specifies the model nugget that will be updated or regenerated each time the stream is updated in the repository (typically as part of a scheduled job). The model must be located on the scoring branch. While multiple models may exist on the scoring branch, only one can be designated. Note that when the stream is initially created this may effectively be a placeholder model that is updated or regenerated as new data is available.

Deploy as stream. Click this option if you want to use the stream with IBM SPSS Modeler Advantage or IBM SPSS Collaboration and Deployment Services.

Check. Click this button to check whether this is a valid stream for deployment. All streams must have a designated scoring node before they can be deployed. Error messages are displayed if these conditions are not satisfied.

Store. Deploys the stream if it is valid. If not, an error message is displayed. Click the Fix button, correct the error, and try again.

Preview Stream Description. Enables you to view the contents of the stream description that IBM SPSS Modeler creates for the stream. See [Stream descriptions](#) for more information.

Note: The Association Rules, STP, and TCM modeling nodes do not support the Model Evaluation or Champion Challenger steps in IBM SPSS Collaboration and Deployment Services.

Stream Deployment Options

The Deployment tab in the Stream Options dialog box allows you to specify options for deploying the stream.

When you deploy a stream, it's stored in the repository as a file with the extension .str.

Deploying a stream allows you to take advantage of the additional functionality available with IBM® SPSS® Collaboration and Deployment Services, such as multi-user access, automated scoring, model refresh, and Champion Challenger analysis.

Note: The Association Rules, STP, and TCM modeling nodes do not support the Model Evaluation or Champion Challenger steps in IBM SPSS Collaboration and Deployment Services.

From the Deployment tab, you can also preview the stream description that IBM SPSS Modeler creates for the stream. See the topic [Stream descriptions](#) for more information.

Deployment type. Choose how you want to deploy the stream. All streams require a designated scoring node before they can be deployed; additional requirements and options depend on the deployment type.

Note: The Association Rules, STP, and TCM modeling nodes do not support the Model Evaluation or Champion Challenger steps in IBM SPSS Collaboration and Deployment Services.

- <none>. The stream will not be deployed to the repository. All options are disabled except stream description preview.
- Scoring Only. The stream is deployed to the repository when you click the Store button. Data can be scored using the node that you designate in the Scoring node field.
- Model Refresh. Same as for Scoring Only but in addition, the model can be updated in the repository using the objects that you designate in the Modeling node and Model nugget fields. Note that automatic model refresh is not supported by default in IBM SPSS Collaboration and Deployment Services, so you must choose this deployment type if you want to use this feature when running a stream from the repository. See [Model Refresh](#) for more information.

Scoring node. Select a graph, output or export node to identify the stream branch to be used for scoring the data. While the stream can actually contain any number of valid branches, models, and terminal nodes, one and only one scoring branch must be designated for purposes of deployment. This is the most basic requirement to deploy any stream.

Scoring Parameters. Allows you to specify parameters that can be modified when the scoring branch is run. See [Scoring and modeling parameters](#) for more information.

Modeling node. For model refresh, specifies the modeling node used to regenerate or update the model in the repository. Must be a modeling node of the same type as that specified for Model nugget.

Model Build Parameters. Allows you to specify parameters that can be modified when the modeling node is run. See [Scoring and modeling parameters](#) for more information.

Model nugget. For model refresh, specifies the model nugget that will be updated or regenerated each time the stream is updated in the repository (typically as part of a scheduled job). The model must be located on the scoring branch. While multiple models may exist on the scoring branch, only one can be designated. Note that when the stream is initially created this may effectively be a placeholder model that is updated or regenerated as new data is available.

Check. Click this button to check whether this is a valid stream for deployment. All streams must have a designated scoring node before they can be deployed. Error messages are displayed if these conditions are not satisfied.

Store. Deploys the stream if it is valid. If not, an error message is displayed. Click the Fix button, correct the error, and try again.

Preview Stream Description. Enables you to view the contents of the stream description that IBM SPSS Modeler creates for the stream. See [Stream descriptions](#) for more information.

Note: The Association Rules, STP, and TCM modeling nodes do not support the Model Evaluation or Champion Challenger steps in IBM SPSS Collaboration and Deployment Services.

- [Scoring and modeling parameters](#)

Scoring and modeling parameters

When deploying a stream to IBM® SPSS® Collaboration and Deployment Services, you can choose which parameters can be viewed or edited each time the model is updated or scored. For example, you might specify maximum and minimum values, or some other value that may be subject to change each time a job is run.

1. To make a parameter visible so it can be viewed or edited after the stream is deployed, select it from the list in the Scoring Parameters dialog box.

The list of available parameters is defined on the Parameters tab in the stream properties dialog box. See the topic [Setting Stream and Session Parameters](#) for more information.

The Scoring Branch

If you are deploying a stream, one branch of the stream must be designated as the **scoring branch** (that is, the one containing the scoring node). When you designate a branch as the scoring branch, that branch is highlighted on the stream canvas, as is the model link to the nugget on the scoring branch. This visual representation is particularly useful in complex streams with multiple branches, where the scoring branch might not be immediately obvious.

Note: Only one stream branch can be designated as the scoring branch.

If the stream already had a scoring branch defined, the newly-designated branch replaces it as the scoring branch. You can set the color of the scoring branch indication by means of a Custom Color option. See the topic [Setting display options](#) for more information.

You can show or hide the scoring branch indication by means of the Show/hide stream markup toolbar button.

Figure 1. Show/hide stream markup toolbar button



- [Identifying the Scoring Branch for Deployment](#)
- [Model Refresh](#)
- [How the Refresh Model is Selected](#)
- [Checking a scoring branch for errors](#)

Identifying the Scoring Branch for Deployment

You can designate the scoring branch either from the pop-up menu of a terminal node, or from the Tools menu. If you use the pop-up menu, the scoring node is set automatically in the Deployment tab of the stream properties.

To designate a branch as the scoring branch (pop-up menu)

1. Connect the model nugget to a terminal node (a processing or output node downstream from the nugget).
2. Right-click the terminal node.
3. On the menu, click Use as Scoring Branch.

To designate a branch as the scoring branch (Tools menu)

1. Connect the model nugget to a terminal node (a processing or output node downstream from the nugget).
2. On the main menu, click:
Tools > Stream Properties > Deployment
3. On the Deployment type list, click Scoring Only or Model Refresh as required. See the topic [Stream Deployment Options](#) for more information.
4. Click the Scoring node field and select a terminal node from the list.
5. Click OK.

Model Refresh

Model refresh is the process of rebuilding an existing model in a stream using newer data. The stream itself does not change in the repository. For example, the algorithm type and stream-specific settings remain the same, but the model is retrained on new data, and updated if the new version of the model works better than the old one.

Only one model nugget in a stream can be set to refresh--this is known as the **refresh model**. If you click the Model Refresh option on the Deployment tab of the stream properties (see [Stream Deployment Options](#)), the model nugget that you designate at that time becomes the refresh model. You can also designate a model as the refresh model from the pop-up menu of a model nugget. The nugget must already be on the scoring branch for this to be possible.

If you turn off the "refresh model" status of a nugget, this is equivalent to setting the deployment type of the stream to Scoring Only, and the Deployment tab of the stream properties dialog box is updated accordingly. You can turn this status on and off by means of the Use as Refresh Model option on the pop-up menu of the nugget on the current scoring branch.

Removing the model link of a nugget on the scoring branch also removes the "refresh model" status of the nugget. You can undo removal of the model link by means of the Edit menu or the toolbar; doing so also reinstates the "refresh model" status of the nugget.

How the Refresh Model is Selected

As well as the scoring branch, the link to the refresh model is also highlighted in the stream. The model nugget chosen as the refresh model, and therefore the link that is highlighted, depends on how many nuggets are in the stream.

Single Model in Stream

If a single linked model nugget is on the scoring branch when it is identified as such, that nugget becomes the refresh model for the stream.

Multiple Models in Stream

If there is more than one linked nugget in the stream, the refresh model is chosen as follows.

If a model nugget has been defined in the Deployment tab of the stream properties dialog box and is also in the stream, then that nugget becomes the refresh model.

If no nugget has been defined in the Deployment tab, or if one has been defined but is not on the scoring branch, then the nugget closest to the terminal node becomes the refresh model.

If you subsequently deselect all model links as refresh links, only the scoring branch is highlighted, not the links. The deployment type is set to Scoring Only.

Note: You can choose to set one of the links to Replace status, but not the other one. In this case, the model nugget chosen as the refresh model is the one that has a refresh link and which is closest to the terminal node when the scoring branch is designated.

No Models in Stream

If there are no models in the stream, or only models with no model links, the deployment type is set to Scoring Only.

Checking a scoring branch for errors

When you designate the scoring branch, it is checked for errors.

If an error is found, the scoring branch is highlighted in the scoring branch error color, and an error message is displayed. You can set the error color by means of a Custom Color option. See the topic [Setting display options](#) for more information.

If an error is found, proceed as follows:

1. Correct the error according to the contents of the error message.
2. On the main menu, click:
Tools > Stream Properties > Deployment
and click Check.
3. If necessary, repeat this process until no errors are found.

Saving streams to IBM Cloud Pak for Data

You can save streams to an IBM Cloud Pak for Data server where you can open them as flows and take advantage of the many features available in Cloud Pak for Data.

Cloud Pak for Data simplifies how you collect, organize, and analyze data and infuse AI across your business. Seamlessly integrate a broad range of data and AI software, leading governance and security, and a unified experience for collaboration. For more information, see <https://www.ibm.com/products/cloud-pak-for-data>. Contact your IBM representative if you're interested in Cloud Pak for Data.

To configure the server connection

These instructions are based on Firefox. Steps may vary based on which web browser you're using.

1. Open Firefox, paste your Cloud Pak for Data URL in the address bar, and press Enter.

You can obtain the URL by opening the Cloud Pak for Data web client and looking at the URL in your browser address bar (it's everything up to and not including /zen). By default, the URL takes the form `https://<namespace>-cpd-<namespace>.apps.<cluster-subdomain>`

where:

- `<namespace>` is your namespace (such as `ab123`).
- `<cluster-subdomain>` is the subdomain name of your cluster (such as `ocp454x86-aqt-mycompany.com`)

In this example, the URL is `https://ab123-cpd-ab123.apps.ocp454x86-aqt-mycompany.com`

For more information about the Cloud Pak for Data URL, see [the Cloud Pak for Data documentation](#).

2. Click the yellow padlock icon to the left of the address bar, then click the arrow beside Connection not secure.

3. Click More Information. The Page Info dialog opens.

4. In the Page Info dialog, go to Security and click View Certificate.

5. Scroll down to Miscellaneous and download the PEM certification file.

6. From the command line, import the certification file to your IBM® SPSS® Modeler client directory as follows:

On Windows:

```
"%JAVA_HOME%\bin\keytool.exe -importcert -trustcacerts -file <Certification file absolute path> -alias <alias> -keystore "<Modeler client installation path>"\jre\lib\security\cacerts
```

On macOS:

```
"%JAVA_HOME%/bin/keytool -importcert -trustcacerts -file <Certification file absolute path> -alias <alias> -keystore "<Modeler client installation path>"/jre/lib/security/cacerts
```

When prompted, enter `changeit` for the keystore password and enter `yes` to trust the certification.

7. On the main menu in IBM SPSS Modeler, click:

Tools > IBM Cloud Pak for Data Server > IBM Cloud Pak for Data Server Connections...

8. Specify connection settings to the server. Use the same URL you used in step 1. If you're not sure which username or password to use, contact your local system administrator. If you select the API Key login method, use the following method to obtain the URL:

- Log on to the Cloud Pak for Data cluster and run the command `oc get routes -n ibm-common-services`, then copy the `HOST/PORT` value for `cp-console`. In the following example, the URL is `https://cp-console.apps.wml46x04.cp.myserver.com`.

NAME	HOST/PORT	PATH	SERVICES	PORT
TERMINATION	WILDCARD			
cp-console	cp-console.apps.wml46x04.cp.myserver.com		icp-management-ingress	https
reencrypt/Redirect	None			
cp-proxy	cp-proxy.apps.wml46x04.cp.myserver.com		nginx-ingress-controller	https
passthrough/Redirect	None			

9. Click Connect to connect to the server.

To save streams to Cloud Pak for Data

1. On the main menu, click:

File...>Store as Flow.... Streams are referred to as *flows* after they're migrated to Cloud Pak for Data.

Alternatively, you can also right-click in the streams manager pane and select Store as Flow...

2. Choose the Cloud Pak for Data Server you configured previously. Then select the project to which you want to save the flow.

Tip: You can also select multiple streams to save at one time. On the main menu, click Tools...>IBM Cloud Pak for Data Server...>Bulk uploading streams to IBM Cloud Pak for Data Server. You'll then be prompted to select streams from your local file system, select a server, and select a project.

Exporting to external applications

- [About Exporting to External Applications](#)
- [Opening a Stream in IBM SPSS Modeler Advantage](#)

About Exporting to External Applications

IBM® SPSS® Modeler provides a number of mechanisms to export the entire data mining process to external applications, so that the work you do to prepare data and build models can be used to your advantage outside of IBM SPSS Modeler as well.

The previous section showed how you can deploy streams to an IBM SPSS Collaboration and Deployment Services repository to take advantage of its multi-user access, job scheduling and other features. In a similar way, IBM SPSS Modeler streams can also be used in conjunction with:

- IBM SPSS Modeler Advantage
- Applications that can import and export files in PMML format

For more information about using streams with IBM SPSS Modeler Advantage, see [Opening a Stream in IBM SPSS Modeler Advantage](#).

For information on exporting and importing models as PMML files, making it possible to share models with any other applications that support this format, see [Importing and exporting models as PMML](#).

Opening a Stream in IBM SPSS Modeler Advantage

IBM® SPSS® Modeler streams can be used in conjunction with the thin-client application IBM SPSS Modeler Advantage. While it is possible to create customized applications entirely within IBM SPSS Modeler Advantage, you can also use a stream already created in IBM SPSS Modeler as the basis of an application workflow.

To open a stream in IBM SPSS Modeler Advantage:

1. Deploy the stream in the IBM SPSS Collaboration and Deployment Services repository, being sure to click the Deploy as stream option. See the topic [Deploying streams](#) for more information.
2. Click the Open in IBM SPSS Modeler Advantage toolbar button, or from the main menu click:

File...>Open in IBM SPSS Modeler Advantage

1. Specify connection settings to the repository if necessary. See the topic [Connecting to the Repository](#) for more information. For specific port, password, and other connection details, contact your local system administrator.

Note: The repository server must also have the IBM SPSS Modeler Advantage software installed.

1. In the Repository: Store dialog, choose the folder where you want to store the object, specify any other information you want to record, and click the Store button. See the topic [Setting Object Properties](#) for more information.

Doing so launches IBM SPSS Modeler Advantage with the stream already open. The stream is closed in IBM SPSS Modeler.

Projects and reports

- [Introduction to Projects](#)
 - [Building a Project](#)
 - [Generating a Report](#)
-

Introduction to Projects

A **project** is a group of files related to a data mining task. Projects include data streams, graphs, generated models, reports, and anything else that you have created in IBM® SPSS® Modeler. At first glance, it may seem that IBM SPSS Modeler projects are simply a way to organize output, but they are actually capable of much more. Using projects, you can:

- Annotate each object in the project file.
- Use the CRISP-DM methodology to guide your data mining efforts. Projects also contain a CRISP-DM Help system that provides details and real-world examples on data mining with CRISP-DM.
- Add non-IBM SPSS Modeler objects to the project, such as a PowerPoint slide show used to present your data mining goals or white papers on the algorithms that you plan to use.
- Produce both comprehensive and simple update reports based on your annotations. These reports can be generated in HTML for easy publishing on your organization's intranet.

Note: If the project pane is not visible in the IBM SPSS Modeler window, click Project on the View menu.

Objects that you add to a project can be viewed in two ways: **Classes view** and **CRISP-DM view**. Anything that you add to a project is added to both views, and you can toggle between views to create the organization that works best.

- [CRISP-DM View](#)
- [Classes View](#)

Related information

- [CRISP-DM View](#)
 - [Classes View](#)
 - [Building a Project](#)
-

CRISP-DM View

By supporting the Cross-Industry Standard Process for Data Mining (CRISP-DM), IBM® SPSS® Modeler projects provide an industry-proven and non-proprietary way of organizing the pieces of your data mining efforts. CRISP-DM uses six phases to describe the process from start (gathering business requirements) to finish (deploying your results). Even though some phases do not typically involve work in IBM SPSS Modeler, the project pane includes all six phases so that you have a central location for storing and tracking all materials associated with the project. For example, the Business Understanding phase typically involves gathering requirements and meeting with colleagues to determine goals rather than working with data in IBM SPSS Modeler. The project pane allows you to store your notes from such meetings in the *Business Understanding* folder for future reference and inclusion in reports.

The CRISP-DM view in the project pane is also equipped with its own Help system to guide you through the data mining life cycle. From IBM SPSS Modeler, this help can be accessed by clicking CRISP-DM Help on the Help menu.

Note: If the project pane is not visible in the window, click Project on the View menu.

- [Setting the Default Project Phase](#)

Related information

- [Introduction to Projects](#)
- [Classes View](#)
- [Building a Project](#)

- [Setting the Default Project Phase](#)
-

Setting the Default Project Phase

Objects added to a project are added to a default phase of CRISP-DM. This means that you need to organize objects manually according to the data mining phase in which you used them. It is wise to set the default folder to the phase in which you are currently working.

To select which phase to use as your default:

1. In CRISP-DM view, right-click the folder for the phase to set as the default.
2. On the menu, click Set as Default.

The default folder is displayed in bold type.

Related information

- [CRISP-DM View](#)
-

Classes View

The Classes view in the project pane organizes your work in IBM® SPSS® Modeler categorically by the types of objects created. Saved objects can be added to any of the following categories:

- Streams
- Nodes
- Models
- Tables, graphs, reports
- Other (non-IBM SPSS Modeler files, such as slide shows or white papers relevant to your data mining work)

Adding objects to the Classes view also adds them to the default phase folder in the CRISP-DM view.

Note: If the project pane is not visible in the window, click Project on the View menu.

Related information

- [Introduction to Projects](#)
 - [Building a Project](#)
 - [CRISP-DM View](#)
 - [Setting the Default Project Phase](#)
-

Building a Project

A project is essentially a file containing references to all of the files that you associate with the project. This means that project items are saved both individually and as a reference in the project file (.cpj). Because of this referential structure, note the following:

- Project items must first be saved individually before being added to a project. If an item is unsaved, you will be prompted to save it before adding it to the current project.
 - Objects that are updated individually, such as streams, are also updated in the project file.
 - Manually moving or deleting objects (such as streams, nodes, and output objects) from the file system will render links in the project file invalid.
- [Creating a New Project](#)
 - [Adding to a Project](#)
 - [Transferring Projects to the IBM SPSS Collaboration and Deployment Services Repository](#)
 - [Setting Project Properties](#)
 - [Annotating a Project](#)
 - [Object Properties](#)
 - [Closing a Project](#)

Related information

- [Introduction to Projects](#)
 - [Adding to a Project](#)
 - [Creating a New Project](#)
 - [Closing a Project](#)
-

Creating a New Project

New projects are easy to create in the IBM® SPSS® Modeler window. You can either start building one, if none is open, or you can close an existing project and start from scratch.

On the main menu, click:

File > Project > New Project...

Related information

- [Building a Project](#)
-

Adding to a Project

Once you have created or opened a project, you can add objects, such as data streams, nodes, and reports, using several methods.

Adding Objects from the Managers

Using the managers in the upper right corner of the IBM® SPSS® Modeler window, you can add streams or output.

1. Select an object, such as a table or a stream, from one of the manager tabs.
2. Right-click, and click Add to Project.
If the object has been previously saved, it will automatically be added to the appropriate objects folder (in Classes view) or to the default phase folder (in CRISP-DM view).
3. Alternatively, you can drag and drop objects from the managers to the project pane.

Note: You may be asked to save the object first. When saving, be sure that Add file to project is selected in the Save dialog box. This will automatically add the object to the project after you save it.

Adding Nodes from the Canvas

You can add individual nodes from the stream canvas by using the Save dialog box.

1. Select a node on the canvas.
2. Right-click, and click Save Node. Alternatively, on the main menu click:
Edit > Node > Save Node...
3. In the Save dialog box, select Add file to project.
4. Create a name for the node and click Save.

This saves the file and adds it to the project. Nodes are added to the *Nodes* folder in Classes view and to the default phase folder in CRISP-DM view.

Adding External Files

You can add a wide variety of non-IBM SPSS Modeler objects to a project. This is useful when you are managing the entire data mining process within IBM SPSS Modeler. For example, you can store links to data, notes, presentations, and graphics in a project. In CRISP-DM view, external files can be added to the folder of your choice. In Classes view, external files can be saved only to the *Other* folder.

To add external files to a project:

1. Drag files from the desktop to the project.
or
2. Right-click the target folder in CRISP-DM or Classes view.
3. On the menu, click Add to Folder.
4. Select a file in the dialog box and click Open.

This will add a reference to the selected object inside IBM SPSS Modeler projects.

Related information

- [Setting the Default Project Phase](#)
 - [Building a Project](#)
-

Transferring Projects to the IBM SPSS Collaboration and Deployment Services Repository

You can transfer an entire project, including all component files, to the IBM® SPSS® Collaboration and Deployment Services Repository in one step. Any objects that are already in the target location will not be moved. This feature also works in reverse: you can transfer entire projects from the IBM SPSS Collaboration and Deployment Services Repository to your local file system.

Transferring a Project

Make sure that the project you want to transfer is open in the project pane.

To transfer a project:

1. Right-click the root project folder and click Transfer Project.
 2. If prompted, log in to IBM SPSS Collaboration and Deployment Services Repository.
 3. Specify the new location for the project and click OK.
-

Setting Project Properties

You can customize a project's contents and documentation by using the project properties dialog box. To access project properties:

1. Right-click an object or folder in the project pane and click Project Properties.
2. Click the Project tab to specify basic project information.

Created. Shows the project's creation date (not editable).

Summary. You can enter a summary for your data mining project that will be displayed in the project report.

Contents. Lists the type and number of components referenced by the project file (not editable).

Save unsaved object as. Specifies whether unsaved objects should be saved to the local file system, or stored in the repository. See the topic [About the IBM SPSS Collaboration and Deployment Services Repository](#) for more information.

Update object references when loading project. Select this option to update the project's references to its components. *Note:* The files added to a project are not saved in the project file itself. Rather, a reference to the files is stored in the project. This means that moving or deleting a file will remove that object from the project.

Related information

- [Building a Project](#)
 - [Annotating a Project](#)
-

Annotating a Project

The project pane provides a number of ways to annotate your data mining efforts. Project-level annotations are often used to track "big-picture" goals and decisions, while folder or node annotations provide additional detail. The Annotations tab provides enough space for you to document project-level details, such as the exclusion of data with irretrievable missing data or promising hypotheses formed during data exploration.

To annotate a project:

1. Select the project folder in either CRISP-DM or Classes view.
 2. Right-click the folder and click Project Properties.
 3. Click the Annotations tab.
 4. Enter keywords and text to describe the project.
- [Folder Properties and Annotations](#)

Related information

- [Building a Project](#)
 - [Setting Project Properties](#)
 - [Folder Properties and Annotations](#)
-

Folder Properties and Annotations

Individual project folders (in both CRISP-DM and Classes view) can be annotated. In CRISP-DM view, this can be an extremely effective way to document your organization's goals for each phase of data mining. For example, using the annotation tool for the *Business Understanding* folder, you can include documentation such as "The business objective for this study is to reduce churn among high-value customers." This text could then be automatically included in the project report by selecting the **Include in report** option.

To annotate a folder:

1. Select a folder in the project pane.
2. Right-click the folder and click **Folder Properties**.

In CRISP-DM view, folders are annotated with a summary of the purpose of each phase as well as guidance on completing the relevant data mining tasks. You can remove or edit any of these annotations.

Name. This area displays the name of the selected field.

Tooltip text. Create custom ToolTips that will be displayed when you hover the mouse pointer over a project folder. This is useful in CRISP-DM view, for example, to provide a quick overview of each phase's goals or to mark the status of a phase, such as "In progress" or "Complete."

Annotation field. Use this field for more lengthy annotations that can be collated in the project report. The CRISP-DM view includes a description of each data mining phase in the annotation, but you should feel free to customize this for your own project.

Include in report. To include the annotation in reports, select **Include in report**.

Related information

- [Building a Project](#)
 - [Annotating a Project](#)
-

Object Properties

You can view object properties and choose whether to include individual objects in the project report. To access object properties:

1. Right-click an object in the project pane.
2. On the menu, click **Object Properties**.

Name. This area lists the name of the saved object.

Path. This area lists the location of the saved object.

Include in report. Select this option to include the object details in a generated report.

Related information

- [Building a Project](#)
 - [Introduction to Projects](#)
-

Closing a Project

When you exit IBM® SPSS® Modeler or open a new project, the existing project file (.cpj) is closed.

Some files associated with the project (such as streams, nodes or graphs) may still be open. If you want to leave these files open, reply **No** to the message ... Do you want to save and close these files?

If you modify and save any associated files after the close of a project, these updated versions will be included in the project the next time you open it. To prevent this behavior, remove the file from the project or save it under a different filename.

Related information

- [Building a Project](#)
-

Generating a Report

One of the most useful features of projects is the ability to generate reports based on the project items and annotations. This is a critical component of effective data mining, as discussed throughout the CRISP-DM methodology. You can generate a report directly into one of several file types or to an output window on the screen for immediate viewing. From there, you can print, save, or view the report in a web browser. You can distribute saved reports to others in your organization.

Reports are often generated from project files several times during the data mining process for distribution to those involved in the project. The report culls information about the objects referenced from the project file as well as any annotations created. You can create reports based on either the Classes view or CRISP-DM view.

To generate a report:

1. Select the project folder in either CRISP-DM or Classes view.
2. Right-click the folder and click Project Report.
3. Specify the report options and click Generate Report.

The options in the report dialog box provide several ways to generate the type of report you need:

Output name. Specify the name of the output window if you choose to send the output of the report to the screen. You can specify a custom name or let IBM® SPSS® Modeler automatically name the window for you.

Output to screen. Select this option to generate and display the report in an output window. Note that you have the option to export the report to various file types from the output window.

Output to file. Select this option to generate and save the report as a file of the type specified in the File type list.

Filename. Specify a filename for the generated report. Files are saved by default to the IBM SPSS Modeler *|bin* directory. Use the ellipsis button (...) to specify a different location.

File type. Available file types are:

- **HTML document.** The report is saved as a single HTML file. If your report contains graphs, they are saved as PNG files and are referenced by the HTML file. When publishing your report on the Internet, make sure to upload both the HTML file and any images it references.
- **Text document.** The report is saved as a single text file. If your report contains graphs, only the filename and path references are included in the report.
- **Microsoft Word document.** The report is saved as a single document, with any graphs embedded directly into the document.
- **Microsoft Excel document.** The report is saved as a single spreadsheet, with any graphs embedded directly into the spreadsheet.
- **Microsoft PowerPoint document.** Each phase is shown on a new slide. Any graphs are embedded directly into the PowerPoint slides.
- **Output object.** When opened in IBM SPSS Modeler, this file (.cou) is the same as the Output to screen option in the Report Format group.

Note: To export to a Microsoft Office file, you must have the corresponding application installed.

Title. Specify a title for the report.

Report structure. Select either CRISP-DM or Classes. CRISP-DM view provides a status report with "big-picture" synopses as well as details about each phase of data mining. Classes view is an object-based view that is more appropriate for internal tracking of data and streams.

Author. The default user name is displayed, but you can change it.

Report includes. Select a method for including objects in the report. Select all folders and objects to include all items added to the project file. You can also include items based on whether Include in Report is selected in the object properties. Alternatively, to check on unreported items, you can choose to include only items marked for exclusion (where Include in Report is not selected).

Select. This option allows you to provide project updates by selecting only recent items in the report. Alternatively, you can track older and perhaps unresolved issues by setting parameters for old items. Select all items to dismiss time as a parameter for the report.

Order by. You can select a combination of the following object characteristics to order them within a folder:

- **Type.** Group objects by type.
- **Name.** Organize objects alphabetically.
- **Added date.** Sort objects using the date they were added to the project.

- [Saving and Exporting Generated Reports](#)

Related information

- [Saving and Exporting Generated Reports](#)

Saving and Exporting Generated Reports

A report generated to the screen is displayed in a new output window. Any graphs included in the report are displayed as in-line images.

Report Terminology

The total number of nodes in each stream is listed within the report. The numbers are shown under the following headings, which use IBM® SPSS® Modeler terminology, not CRISP-DM terminology:

- **Data readers.** Source nodes.
- **Data writers.** Export nodes.
- **Model builders.** Build, or Modeling, nodes.
- **Model appliers.** Generated models, also known as nuggets.
- **Output builders.** Graph or Output nodes.
- **Other.** Any other nodes related to the project. For example, those available on the Field Ops tab or Record Ops tab on the Nodes Palette.

To save a report:

1. On the File menu, click Save.
2. Specify a filename.
The report is saved as an output object.

To export a report:

3. On the File menu, click Export and the file type to which you want to export.
4. Specify a filename.

The report is saved in the format you chose.

You can export to the following file types:

- HTML
- Text
- Microsoft Word
- Microsoft Excel
- Microsoft PowerPoint

Note: To export to a Microsoft Office file, you must have the corresponding application installed.

Use the buttons at the top of the window to:

- Print the report.
- View the report as HTML in an external web browser.

Related information

- [Generating a Report](#)

Customizing IBM® SPSS® Modeler

- [Customizing IBM SPSS Modeler options](#)
- [Setting IBM SPSS Modeler options](#)
- [Customizing the Nodes Palette](#)

Customizing IBM SPSS Modeler options

There are a number of operations you can perform to customize IBM® SPSS® Modeler to your needs. Primarily, this customization consists of setting specific user options such as memory allocation, default directories, and the use of sound and color. You can also customize the Nodes palette located at the bottom of the IBM SPSS Modeler window.

Related information

- [Setting IBM SPSS Modeler options](#)
 - [Customizing the Nodes Palette](#)
-

Setting IBM SPSS Modeler options

There are several ways to customize and set options for IBM® SPSS® Modeler:

- Set system options, such as memory usage and locale, by clicking System Options on the Tools > Options menu.
- Set user options, such as display fonts and colors, by clicking User Options on the Tools > Options menu.
- Specify the location of applications that work with IBM SPSS Modeler by clicking Helper Applications on the Tools > Options menu.
- Specify the default directories used in IBM SPSS Modeler by clicking Set Directory or Set Server Directory on the File menu.

You can also set options that apply to some or all of your streams. See the topic [Setting options for streams](#) for more information.

- [System Options](#)
 - [Setting Default Directories](#)
 - [Setting user options](#)
-

System Options

You can specify the preferred language or locale for IBM® SPSS® Modeler by clicking System Options on the Tools > Options menu. Here you can also set the maximum memory usage for SPSS Modeler, and specify how often to automatically save streams. Note that changes made in this dialog box will not take effect until you restart SPSS Modeler.

Maximum memory. Select to impose a limit in megabytes on IBM SPSS Modeler's memory usage. On some platforms, SPSS Modeler limits its process size to reduce the toll on computers with limited resources or heavy loads. If you are dealing with large amounts of data, this may cause an "out of memory" error. You can ease memory load by specifying a new threshold.

For example, attempting to display a very large decision tree may cause a memory error. In this case, we recommend that you increase the memory to the high value such as 4096Mb. In cases such as these, where you are likely to be processing very large amounts of data, after you increase the memory allowance shut down SPSS Modeler and start it from a command line to ensure the maximum amount of memory is used when processing your data.

To start from a command line (assuming SPSS Modeler is installed in the default location), in a Command Prompt window, enter the following:

```
C:\Program Files\IBM\SPSS\Modeler\18.4.0\bin\modelerclient.exe" -J-Xss4096M
```

Note: The maximum memory threshold you can specify is 58368Mb due a Java constraint.

Use system locale. This option is selected by default and set to English (United States). Deselect to specify another language from the list of available languages and locales.

Stream auto save interval (minutes). Specify how often you want SPSS Modeler to save streams automatically. The maximum is 60 minutes, minimum is 1 minute, and default is 5 minutes.

- [Managing Memory](#)
-

Managing Memory

In addition to the Maximum memory setting specified in the System Options dialog box, there are several ways you can optimize memory usage:

- Adjust the Maximum members for nominal fields option in the stream properties dialog box. This option specifies a maximum number of members for nominal fields after which the measurement level of the field becomes *Typeless*. See the topic [Setting general options for streams](#) for more information.
- Force IBM® SPSS® Modeler to free up memory by clicking in the lower right corner of the window where the memory that IBM SPSS Modeler is using and the amount allocated are displayed (xxMB / xxMB). Clicking this region turns it a darker shade, after which memory allocation figures will drop. Once the region returns to its regular color, IBM SPSS Modeler has freed up all the memory possible.

Setting Default Directories

You can specify the default directory used for file browsers and output by selecting Set Directory or Set Server Directory from the File menu.

- **Set Directory.** You can use this option to set the working directory. The default working directory is based on the installation path of your version of IBM® SPSS® Modeler or from the command line path used to launch IBM SPSS Modeler. In local mode, the working directory is the path used for all client-side operations and output files (if they are referenced with relative paths).
- **Set Server Directory.** The Set Server Directory option on the File menu is enabled whenever there is a remote server connection. Use this option to specify the default directory for all server files and data files specified for input or output. The default server directory is `$CLEO/data`, where `$CLEO` is the directory in which the Server version of IBM SPSS Modeler is installed. Using the command line, you can also override this default by using the `-server_directory` flag with the `modelerclient` command line argument.

Related information

- [Setting IBM SPSS Modeler options](#)

Setting user options

You can set general options for IBM® SPSS® Modeler by selecting User Options from the Tools > Options menu. These options apply to all streams used in IBM SPSS Modeler.

The following types of options can be set by clicking the corresponding tab:

- Notification options, such as model overwriting and error messages.
- Display options, such as graph and background colors.
- Syntax color display options.
- PMML export options used when exporting models to Predictive Model Markup Language (PMML).
- User or author information, such as your name, initials, and e-mail address. This information may be displayed on the Annotations tab for nodes and for other objects that you create.
- Switching between traditional mode and Analytic Server mode.

To set stream-specific options, such as decimal separators, time and data formats, optimization, stream layout, and stream scripts, use the Stream Properties dialog box, available from the File and Tools menus.

- [Setting Notification Options](#)
- [Setting display options](#)
- [Setting Syntax Display Options](#)
- [Setting PMML Export Options](#)
- [Setting User Information](#)
- [Setting the mode](#)

Setting Notification Options

Using the Notifications tab of the User Options dialog box, you can set various options regarding the occurrence and type of warnings and confirmation windows in IBM® SPSS® Modeler. You can also specify the behavior of the Outputs and Models tabs in the managers pane when new output and models are generated.

Show stream execution feedback dialog Select to display a dialog box, that includes a progress indicator, when a stream has been running for three seconds. The dialog box also includes details of the output objects created by the stream.

- Close dialog upon completion By default, the dialog box closes when the stream finishes running. Clear this check box if you want the dialog box to remain visible when the stream finishes.

Warn when a node overwrites a file Select to warn with an error message when node operations overwrite an existing file.

Warn when a node overwrites a database table Select to warn with an error message when node operations overwrite an existing database table.

Sound Notifications

Use the list to specify whether sounds notify you when an event or error occurs. There are a number of sounds available. Use the Play (loudspeaker) button to play a selected sound. Use the ellipsis button (...) to browse for and select a sound.

Note: The .wav files used to create sounds in IBM SPSS Modeler are stored in the /media/sounds directory of your installation.

- Mute all sounds Select to turn off sound notification for all events.

Visual Notifications

The options in this group are used to specify the behavior of the Outputs and Models tabs in the managers pane at the top right of the display when new items are generated. Select New Model or New Output from the list to specify the behavior of the corresponding tab.

The following option is available for New Model:

Replace previous model If selected (default), overwrites an existing model from this stream in the Models tab and on the stream canvas. If this box is unchecked, the model is added to the existing models on the tab and the canvas. Note that this setting is overridden by the model replacement setting on a model link.

The following option is available for New Output:

Warn when outputs exceed [n] Select whether to display a warning when the number of items on the Outputs tab exceeds a prespecified quantity. The default quantity is 20; however, you can change this if needed.

The following options are available in all cases:

Select tab Choose whether to switch to the Outputs or Models tab when the corresponding object is generated while the stream runs.

- Select Always to switch to the corresponding tab in the managers pane.
- Select If generated by current stream to switch to the corresponding tab only for objects generated by the stream currently visible in the canvas.
- Select Never to restrict the software from switching to the corresponding tab to notify you of generated outputs or models.

Flash tab Select whether to flash the Outputs or Models tab in the managers pane when new outputs or models have been generated.

- Select If not selected to flash the corresponding tab (if not already selected) whenever new objects are generated in the managers pane.
- Select Never to restrict the software from flashing the corresponding tab to notify you of generated objects.

Scroll palette to make visible (New Model only). Select whether to automatically scroll the Models tab in the managers pane to make the most recent model visible.

- Select Always to enable scrolling.
- Select If generated by current stream to scroll only for objects generated by the stream currently visible in the canvas.
- Select Never to restrict the software from automatically scrolling the Models tab.

Open window (New Output only). Select whether to automatically open an output window upon generation.

- Select Always to always open a new output window.
- Select If generated by current stream to open a new window for output generated by the stream currently visible in the canvas.
- Select Never to restrict the software from automatically opening new windows for generated output.

To revert to the system default settings for this tab, click Default Values.

Related information

- [Setting user options](#)
- [Setting IBM SPSS Modeler options](#)

Setting display options

Using the Display tab of the User Options dialog box, you can set options for the display of fonts and colors in IBM® SPSS® Modeler.

Show welcome dialog on startup. Select to cause the welcome dialog box to be displayed on startup. The welcome dialog box has options to launch the application examples tutorial, open a demonstration stream or an existing stream or project, or to create a new stream.

Show stream and SuperNode markups. If selected, causes markup (if any) on streams and SuperNodes to be displayed by default. Markup includes stream comments, model links, and highlighting of scoring branches.

Standard Fonts & Colors (effective on restart). Options in this control box are used to specify the IBM SPSS Modeler screen design, color scheme, and the size of the fonts displayed. The options that you select here do not take effect until you close and restart IBM SPSS Modeler.

- Look and feel. Select a standard color scheme and screen design. You can choose from:
 - SPSS Standard, the default design.

- SPSS Classic, a design familiar to users of earlier versions of SPSS Modeler.
- Windows, a Windows design that can be useful for increased contrast in the stream canvas and palettes.
- Analytics Carbon, a modern design with sleek icons and colors.
- Default font size for nodes. Specify a font size to be used in the node palettes and for nodes that are displayed in the stream canvas.
- Specify fixed width font. To select a fixed width font, and associated font Size for use in scripting and CLEM expression controls, select this check box. The default font is **Monospace plain**; click Change... to display a list of other fonts that you can select.

Note: You can set the size of the node icons for a stream on the Layout pane of the Options tab of the stream properties dialog box. From the main menu, choose Tools > Stream Properties > Options > Layout.

Custom Colors. This table lists the currently selected colors that are used for various display items. For each of the items that are listed in the table, you can change the current color by double-clicking the corresponding row in the Color column and selecting a color from the list. To specify a custom color, scroll to the bottom of the list and click the Color... entry.

Chart Category Color Order. This table lists the currently selected colors that are used for display in newly created graphs. The order of the colors reflects the order in which they are used in the chart. For example, if a nominal field used as a color overlay contains four unique values, then only the first four colors that are listed here are used. For each of the items that are listed in the table, you can change the current color by double-clicking the corresponding row in the Color column and selecting a color from the list. To specify a custom color, scroll to the bottom of the list and click the Color... entry. Changes that you make here do not affect previously created graphs.

To revert to the system default settings for this tab, click Default Values.

Setting Syntax Display Options

Using the Syntax tab of the User Options dialog box, you can set options for the font attributes and display colors in scripts that you create in IBM® SPSS® Modeler.

Syntax highlighting. This table lists the currently selected colors used for various syntax items, including both the font and the window in which it is displayed. For each of the items listed in the table you can change the color by clicking the corresponding drop-down list in the row and selecting a color from the list. In addition, for font items, you can choose to add bold and italic emphasis.

Preview. This table shows an example syntax display that uses the colors and font attributes that you select in the Syntax highlighting table. This preview updates as soon as you change any selection.

Click Default Values to revert to the system default settings for this tab.

Related information

- [Setting user options](#)
- [Setting IBM SPSS Modeler options](#)
- [Adding Comments and Annotations to Nodes and Streams](#)

Setting PMML Export Options

On the PMML tab, you can control how IBM® SPSS® Modeler exports models to Predictive Model Markup Language (PMML). See the topic [Importing and exporting models as PMML](#) for more information.

Export PMML. Here you can configure variations of PMML that work best with your target application.

- Select With extensions to allow PMML extensions for special cases where there is no standard PMML equivalent. Note that in most cases this will produce the same result as standard PMML.
- Select As standard PMML... to export PMML that adheres as closely as possible to the PMML standard.

Standard PMML Options. When the As standard PMML... option is selected, you can choose one of two valid ways to export linear and logistic regression models:

- As PMML <GeneralRegression> models
- As PMML <Regression> models

For more information on PMML, see the Data Mining Group website at <http://www.dmg.org>.

Related information

- [Setting user options](#)
- [Setting IBM SPSS Modeler options](#)

- [Importing and exporting models as PMML](#)
- [Model types supporting PMML](#)

Setting User Information

User/Author Information. Information you enter here can be displayed on the Annotations tab of nodes and other objects that you create.

Setting the mode

Modeler Mode Settings. On the Mode tab, you can choose from the following modes:

- Traditional SPSS Modeler mode shows all available nodes and expressions in the user interface.
- Analytic Server mode shows only the nodes and expressions supported by Analytic Server. But note that some nodes and some CLEM expressions will still be displayed even though they're not *fully* supported by Analytic Server. The following table provides general information about which nodes are supported, partially supported, and unsupported by Analytic Server.

See [Supported nodes](#) for further details.

See the [Analytic Server documentation](#) for more information about Analytic Server.

Note that if you configure SPSS® Modeler to show database nodes on the Database Modeling palette, they will not be impacted when switching modes. The database nodes will always be displayed. If you use IBM Db2 for z/OS or IBM Netezza integration, in some cases those nodes may disappear from the Database Modeling palette after switching to Analytic Server mode. If this happens, go to Tools > Options > Helper Applications and reset the check boxes.

Table 1. Node support

Node type (palette name)	Supported by Analytic Server	Partially supported by Analytic Server	Unsupported by Analytic Server
Sources	<ul style="list-style-type: none"> • Analytic Server source node 		<ul style="list-style-type: none"> • Database • Var. File • Fixed File • Statistics File • Data Collection • IBM Cognos • TWC Import • TM1 Import • SAS File • Excel • XML User Input • Sim Gen • Data View • Geospatial • Object Storage • Extension Import • R Import • SNA (Diffusion Analysis and Group Analysis)

Node type (palette name)	Supported by Analytic Server	Partially supported by Analytic Server	Unsupported by Analytic Server
Record Ops	<ul style="list-style-type: none"> • Select • Sort • Balance • Distinct • RFM Aggregate • Append • Streaming TS • Extension Transformation • Streaming TCM 	<ul style="list-style-type: none"> • Sample (Only supports Random% for Simple method. Complex is not supported.) • Merge (only supports join by keys and conditions) • Aggregate (1st quantile, 3rd quantile, and median are not supported by Analytic Server) 	<ul style="list-style-type: none"> • Space-Time-Boxes • CPLEX Optimization
Field Ops	<ul style="list-style-type: none"> • Type • Filter • Derive • Filler • Reclassify • Ensemble • SetToFlag • Restructure • Field Reorder • Reproject • Time Intervals 	<ul style="list-style-type: none"> • Auto Data Prep (only supports transform) • Binning (binning method <code>equalFreq</code> is not supported when Ties setting "Keep in current" is selected) • RFM Analysis (binning method <code>tiles</code> is not supported when Ties setting "Keep in current" is selected) • Partition (not supported by Analytic Server unless a unique field is used to repeatedly assign rows to partitions) 	<ul style="list-style-type: none"> • Anonymize • History • Transpose
Graphs	<ul style="list-style-type: none"> • Plot • Multiplot • Time Plot • Distribution • Histogram • Collection • Web • Evaluation • Map Visualisation • E-Plot (Beta) • t-SNE 	<ul style="list-style-type: none"> • Graphboard (only supports the aggregation mode function for fields having a measurement level of discrete, nominal, ordinal, or flag) 	

Node type (palette name)	Supported by Analytic Server	Partially supported by Analytic Server	Unsupported by Analytic Server
Modeling	<ul style="list-style-type: none"> • Time Series • TCM • Isotonic-AS • Random Trees • Tree-AS • Linear-AS • GLE • LSVM • STP • TwoStep-AS • Association Rules • XGBoost-AS • K-Means-AS 	<ul style="list-style-type: none"> • Auto Classifier • Auto Numeric (the two nodes only support split, and a field with a split role must be supplied when using the Auto Classifier option Run on Analytic Server (splits enabled)) • Extension (the R syntax building model is not supported by Analytic Server) • The following nodes only support split and PSM: <ul style="list-style-type: none"> ◦ C&R Tree ◦ Linear ◦ Neural Net ◦ CHAID ◦ Quest 	<ul style="list-style-type: none"> • Auto cluster • Decision List • C5.0 • Regression • PCA/Factor • Feature Selection • Discriminant • Logistic • GenLin • GLMM • Bayes Net • Apriori • Carma • Sequence • K-Means • Kohonen • TwoStep • Anomaly • KNN • R • Random Forest • The following nodes have asl but just read-write asl: <ul style="list-style-type: none"> ◦ Cox ◦ SVM ◦ SLRM
Output	<ul style="list-style-type: none"> • Table • Matrix • Analysis • Data Audit • Transformation • Statistics • Means • Report • Set Globals 	<ul style="list-style-type: none"> • Extension Output • R Syntax Output 	<ul style="list-style-type: none"> • Sim Eval • Sim Fit • R Output
Export	<ul style="list-style-type: none"> • Analytic Server export node 	<ul style="list-style-type: none"> • Extension Export • R Syntax Export 	<ul style="list-style-type: none"> • Database • Flat File • Statistics Export • Data Collection • Excel • IBM Cognos Export • TM1 Export • SAS • XML Export • Object Storage • R Export
IBM SPSS Statistics			<ul style="list-style-type: none"> • Statistics File • Statistics Transform • Statistics Model • Statistics Output • Statistics Export
IBM SPSS Text Analytics	<ul style="list-style-type: none"> • Text Link Analysis • Text Mining • Language node 		<ul style="list-style-type: none"> • File List • Web Feed • Text Link Analysis • Translate • Text Mining • File Viewer

Node type (palette name)	Supported by Analytic Server	Partially supported by Analytic Server	Unsupported by Analytic Server
Python			<ul style="list-style-type: none"> SMOTE One-Class SVM XGBoost Tree XGBoost Linear t-SNE Random Forest HDBSCAN
Spark	All supported		

Customizing the Nodes Palette

Streams are built using nodes. The Nodes Palette at the bottom of the IBM® SPSS® Modeler window contains all of the nodes it is possible to use in stream building. See the topic [Nodes palette](#) for more information.

You can reorganize the Nodes Palette in two ways:

- Customize the Palette Manager. See the topic [Customizing the Palette Manager](#) for more information.
- Change how palette tabs that contain subpalettes are displayed on the Nodes Palette. See the topic [Creating a Subpalette](#) for more information.
- [Customizing the Palette Manager](#)
- [Changing a Palette Tab View](#)

Related information

- [Customizing the Palette Manager](#)
- [Creating a Palette Tab](#)
- [Displaying Palette Tabs on the Nodes Palette](#)
- [Displaying Subpalettes on a Palette Tab](#)
- [Creating a Subpalette](#)
- [Changing a Palette Tab View](#)
- [Customizing IBM SPSS Modeler options](#)
- [Setting IBM SPSS Modeler options](#)

Customizing the Palette Manager

The Palette Manager can be customized to accommodate your usage of IBM® SPSS® Modeler. For example, if you frequently analyze time-series data from a database, you might want to be sure that the Database source node, the Time intervals node, the Time Series node, and the Time Plot graph node are available together from a unique palette tab. The Palette Manager enables you to easily make these adjustments by creating your custom palette tabs in the Nodes Palette.

The Palette Manager enables you to carry out various tasks:

- Control which palette tabs are shown on the Nodes Palette below the stream canvas.
- Change the order in which palette tabs are shown on the Nodes Palette.
- Create and edit your own palette tabs and any associated subpalettes.
- Edit the default node selections on your tabs.

To access the Palette Manager, on the Tools menu, click Manage Palettes.

Palette Name. Each available palette tab, whether shown on the Nodes Palette or not, is listed. This includes any palette tabs that you have created. See the topic [Creating a Palette Tab](#) for more information.

No. of nodes. The number of nodes displayed on each palette tab. A high number here means you may find it more convenient to create subpalettes to divide up the nodes on the tab. See the topic [Creating a Subpalette](#) for more information.

Shown? Select this field to display the palette tab on the Nodes Palette. See the topic [Displaying Palette Tabs on the Nodes Palette](#) for more information.

Sub Palettes. To select subpalettes for display on a palette tab, highlight the required Palette Name and click this button to display the Sub Palettes dialog box. See the topic [Creating a Subpalette](#) for more information.

Restore Defaults. To completely remove all changes and additions you have made to the palettes and subpalettes and return to the default palette settings, click this button.

- [Creating a Palette Tab](#)
 - [Displaying Palette Tabs on the Nodes Palette](#)
 - [Displaying Subpalettes on a Palette Tab](#)
 - [Creating a Subpalette](#)
-

Creating a Palette Tab

To create a custom palette tab:

1. From the Tools menu, open the Palette Manager.
2. To the right of the *Shown?* column, click the Add Palette button; the Create/Edit Palette dialog box is displayed.
3. Type in a unique Palette name.
4. In the Nodes available area, select the node to be added to the palette tab.
5. Click the Add Node right-arrow button to move the highlighted node to the Selected nodes area. Repeat until you have added all the nodes you want.

After you have added all of the required nodes, you can change the order in which they are displayed on the palette tab:

6. Use the simple arrow buttons to move a node up or down one row.
7. Use the line-arrow buttons to move a node to the bottom or top of the list.
8. To remove a node from a palette, highlight the node and click the Delete button to the right of the Selected nodes area.

Related information

- [Customizing the Nodes Palette](#)
 - [Customizing the Palette Manager](#)
 - [Displaying Palette Tabs on the Nodes Palette](#)
 - [Displaying Subpalettes on a Palette Tab](#)
 - [Creating a Subpalette](#)
 - [Changing a Palette Tab View](#)
-

Displaying Palette Tabs on the Nodes Palette

There may be options available within IBM® SPSS® Modeler that you never use; in this case, you can use the Palette Manager to hide the tabs containing these nodes.

To select which tabs are to be shown on the Nodes Palette:

1. From the Tools menu, open the Palette Manager.
2. Using the check boxes in the *Shown?* column, select whether to include or hide each palette tab.

To permanently remove a palette tab from the Nodes Palette, highlight the node and click the Delete button to the right of the *Shown?* column. Once deleted, a palette tab cannot be recovered.

Note: You cannot delete the default palette tabs supplied with IBM SPSS Modeler, except for the Favorites tab.

Changing the display order on the Nodes Palette

After you have selected which palette tabs you want to display, you can change the order in which they are displayed on the Nodes Palette:

1. Use the simple arrow buttons to move a palette tab up or down one row. Moving them up moves them to the left of the Nodes Palette, and vice versa.
2. Use the line-arrow buttons to move a palette tab to the bottom or top of the list. Those at the top of the list will be shown on the left of the Nodes Palette.

Related information

- [Customizing the Nodes Palette](#)
 - [Customizing the Palette Manager](#)
 - [Creating a Palette Tab](#)
 - [Displaying Subpalettes on a Palette Tab](#)
 - [Creating a Subpalette](#)
-

- [Changing a Palette Tab View](#)
-

Displaying Subpalettes on a Palette Tab

In the same way that you can control which palette tabs are displayed on the Nodes Palette, you can control which subpalettes are available from their parent palette tab.

To select subpalettes for display on a palette tab:

1. From the Tools menu, open the Palette Manager.
2. Select the palette that you require.
3. Click the Sub Palettes button; the Sub Palettes dialog box is displayed.
4. Using the check boxes in the *Shown?* column, select whether to include each subpalette on the palette tab. The All subpalette is always shown and cannot be deleted.
5. To permanently remove a subpalette from the palette tab, highlight the subpalette and click the Delete button to the right of the *Shown?* column.

Note: You cannot delete the default subpalettes supplied with the Modeling palette tab.

Changing the display order on the Palette Tab

After you have selected which subpalettes you want to display, you can change the order in which they are displayed on the parent palette tab:

1. Use the simple arrow buttons to move a subpalette up or down one row.
2. Use the line-arrow buttons to move a subpalette to the bottom or top of the list.

The subpalettes you create are displayed on the Nodes Palette when you select their parent palette tab. See the topic [Changing a Palette Tab View](#) for more information.

Related information

- [Customizing the Nodes Palette](#)
 - [Customizing the Palette Manager](#)
 - [Creating a Palette Tab](#)
 - [Displaying Palette Tabs on the Nodes Palette](#)
 - [Creating a Subpalette](#)
 - [Changing a Palette Tab View](#)
-

Creating a Subpalette

Because you can add any existing node to the custom palette tabs that you create, it is possible that you will select more nodes than can be easily displayed on screen without scrolling. To prevent having to scroll, you can create subpalettes into which you place the nodes you chose for the palette tab. For example, if you created a palette tab that contains the nodes you use most frequently for creating your streams, you could create four subpalettes that break the selections down by source node, field operations, modeling, and output.

Note: You can only select subpalette nodes from those added to the parent palette tab.

To create a subpalette:

1. From the Tools menu, open the Palette Manager.
2. Select the palette to which you want to add subpalettes.
3. Click the Sub Palettes button; the Sub Palettes dialog box is displayed.
4. To the right of the *Shown?* column, click the Add Sub Palette button; the Create/Edit Sub Palette dialog box is displayed.
5. Type in a unique Sub palette name.
6. In the Nodes available area, select the node to be added to the subpalette.
7. Click the Add Node right-arrow button to move a selected node to the Selected nodes area.
8. When you have added the required nodes, click OK to return to the Sub Palettes dialog box.

The subpalettes you create are displayed on the Nodes Palette when you select their parent palette tab. See the topic [Changing a Palette Tab View](#) for more information.

Related information

- [Customizing the Nodes Palette](#)

- [Customizing the Palette Manager](#)
 - [Creating a Palette Tab](#)
 - [Displaying Palette Tabs on the Nodes Palette](#)
 - [Displaying Subpalettes on a Palette Tab](#)
 - [Changing a Palette Tab View](#)
-

Changing a Palette Tab View

Due to the large number of nodes available in IBM® SPSS® Modeler, they may not all be visible on smaller screens without scrolling to the left or right of the Nodes Palette; this is especially noticeable on the Modeling palette tab. To reduce the need to scroll, you can choose to display only the nodes contained in a subpalette (where available). See the topic [Creating a Subpalette](#) for more information.

To change the nodes shown on a palette tab, select the palette tab and then, from the menu on the left, select to display either all nodes, or just those in a specific subpalette.

Related information

- [Customizing the Nodes Palette](#)
 - [Customizing the Palette Manager](#)
 - [Creating a Palette Tab](#)
 - [Displaying Palette Tabs on the Nodes Palette](#)
 - [Displaying Subpalettes on a Palette Tab](#)
 - [Creating a Subpalette](#)
-

Performance considerations for streams and nodes

You can design your streams to maximize performance by arranging the nodes in the most efficient configuration, by enabling node caches when appropriate, and by paying attention to other considerations as detailed in this section.

Aside from the considerations discussed here, additional and more substantial performance improvements can typically be gained by making effective use of your database, particularly through SQL optimization.

- [Order of Nodes](#)
 - [Node Caches](#)
 - [Performance: Process Nodes](#)
 - [Performance: Modeling Nodes](#)
 - [Performance: CLEM Expressions](#)
-

Order of Nodes

Even when you are not using SQL optimization, the order of nodes in a stream can affect performance. The general goal is to minimize downstream processing; therefore, when you have nodes that reduce the amount of data, place them near the beginning of the stream. IBM® SPSS® Modeler Server can apply some reordering rules automatically during compilation to bring forward certain nodes when it can be proven safe to do so. (This feature is enabled by default. Check with your system administrator to make sure it is enabled in your installation.)

When using SQL optimization, you want to maximize its availability and efficiency. Since optimization halts when the stream contains an operation that cannot be performed in the database, it is best to group SQL-optimized operations together at the beginning of the stream. This strategy keeps more of the processing in the database, so less data is carried into IBM SPSS Modeler.

The following operations can be done in most databases. Try to group them at the *beginning* of the stream:

- Merge by key (join)
- Select
- Aggregate
- Sort
- Sample
- Append
- Distinct operations in *include* mode, in which all fields are selected
- Filler operations
- Basic derive operations using standard arithmetic or string manipulation (depending on which operations are supported by the database)
- Set-to-flag

The following operations cannot be performed in most databases. They should be placed in the stream *after* the operations in the preceding list:

- Operations on any nondatabase data, such as flat files
 - Merge by order
 - Balance
 - Distinct operations in *discard* mode or where only a subset of fields are selected as distinct
 - Any operation that requires accessing data from records other than the one being processed
 - State and count field derivations
 - History node operations
 - Operations involving "@" (time-series) functions
 - Type-checking modes *Warn* and *Abort*
 - Model construction, application, and analysis
- Note:* Decision trees, rulesets, linear regression, and factor-generated models can generate SQL and can therefore be pushed back to the database.
- Data output to anywhere other than the same database that is processing the data

Related information

- [Node Caches](#)
-

Node Caches

To optimize stream running, you can set up a *cache* on any nonterminal node. When you set up a cache on a node, the cache is filled with the data that passes through the node the next time you run the data stream. From then on, the data is read from the cache (which is stored on disk in a temporary directory) rather than from the data source.

Caching is most useful following a time-consuming operation such as a sort, merge, or aggregation. For example, suppose that you have a source node set to read sales data from a database and an Aggregate node that summarizes sales by location. You can set up a cache on the Aggregate node rather than on the source node because you want the cache to store the aggregated data rather than the entire data set.

Note: Caching at source nodes, which simply stores a copy of the original data as it is read into IBM® SPSS® Modeler, will not improve performance in most circumstances.

Nodes with caching enabled are displayed with a small document icon at the top right corner. When the data is cached at the node, the document icon is green.

To Enable a Cache

1. On the stream canvas, right-click the node and click Cache on the menu.
2. On the caching submenu, click Enable.
3. You can turn the cache off by right-clicking the node and clicking Disable on the caching submenu.

Caching Nodes in a Database

For streams run in a database, data can be cached midstream to a temporary table in the database rather than the file system. When combined with SQL optimization, this may result in significant gains in performance. For example, the output from a stream that merges multiple tables to create a data mining view may be cached and reused as needed. By automatically generating SQL for all downstream nodes, performance can be further improved.

To take advantage of database caching, both SQL optimization and database caching must be enabled. Note that Server optimization settings override those on the Client. See the topic [Setting optimization options for streams](#) for more information.

With database caching enabled, simply right-click any nonterminal node to cache data at that point, and the cache will be created automatically directly in the database the next time the stream is run. If database caching or SQL optimization is not enabled, the cache will be written to the file system instead.

Note: The following databases support temporary tables for the purpose of caching: Db2, Oracle, SQL Server, and Teradata. Other databases, such as Netezza, will use a normal table for database caching. The SQL code can be customized for specific databases - contact Services for assistance.

Related information

- [Caching options for nodes](#)
- [Order of Nodes](#)

Performance: Process Nodes

Sort. The Sort node must read the entire input data set before it can be sorted. The data is stored in memory up to some limit, and the excess is spilled to disk. The sorting algorithm is a combination algorithm: data is read into memory up to the limit and sorted using a fast hybrid quick-sort algorithm. If all the data fits in memory, then the sort is complete. Otherwise, a merge-sort algorithm is applied. The sorted data is written to file and the next chunk of data is read into memory, sorted, and written to disk. This is repeated until all the data has been read; then the sorted chunks are merged. Merging may require repeated passes over the data stored on disk. At peak usage, the Sort node will have two complete copies of the data set on disk: sorted and unsorted.

The overall running time of the algorithm is on the order of $N \log(N)$, where N is the number of records. Sorting in memory is faster than merging from disk, so the actual running time can be reduced by allocating more memory to the sort. The algorithm allocates to itself a fraction of physical RAM controlled by the IBM® SPSS® Modeler Server configuration option *Memory usage multiplier*. To increase the memory used for sorting, provide more physical RAM or increase this value. Note that when the proportion of memory used exceeds the working set of the process so that part of the memory is paged to disk, performance degrades because the memory-access pattern of the in-memory sort algorithm is random and can cause excessive paging. The sort algorithm is used by several nodes other than the Sort node, but the same performance rules apply.

Binning. The Binning node reads the entire input data set to compute the bin boundaries, before it allocates records to bins. The data set is cached while the boundaries are computed; then it is rescanned for allocation. When the binning method is *fixed-width* or *mean+standard deviation*, the data set is cached directly to disk. These methods have a linear running time and require enough disk space to store the entire data set. When the binning method is *ranks* or *tiles*, the data set is sorted using the sort algorithm described earlier, and the sorted data set is used as the cache. Sorting gives these methods a running time of $M \log(N)$, where M is the number of binned fields and N is the number of records; it requires disk space equal to twice the data set size.

Generating a Derive node based on generated bins will improve performance in subsequent passes. Derive operations are much faster than binning.

Merge by Key (Join). The Merge node, when the merge method is *keys* (equivalent to a database join), sorts each of its input data sets by the key fields. This part of the procedure has a running time of $M \log(N)$, where M is the number of inputs and N is the number of records in the largest input; it requires sufficient disk space to store all of its input data sets plus a second copy of the largest data set. The running time of the merge itself is proportional to the size of the output data set, which depends on the frequency of matching keys. In the worst case, where the output is the Cartesian product of the inputs, the running time may approach MN . This is rare—most joins have many fewer matching keys. If one data set is relatively larger than the other(s), or if the incoming data is already sorted by a key field, then you can improve the performance of this node using the Optimization tab.

Aggregate. When the *Keys are contiguous* option is not set, this node reads (but does not store) its entire input data set before it produces any aggregated output. In the more extreme situations, where the size of the aggregated data reaches a limit (determined by the IBM SPSS Modeler Server configuration option *Memory usage multiplier*), the remainder of the data set is sorted and processed as if the *Keys are contiguous* option were set. When this option is set, no data is stored because the aggregated output records are produced as the input data is read.

Distinct. The Distinct node stores all of the unique key fields in the input data set; in cases where all fields are key fields and all records are unique it stores the entire data set. By default the Distinct node sorts the data on the key fields and then selects (or discards) the first distinct record from each group. For smaller data sets with a low number of distinct keys, or those that have been pre-sorted, you can choose options to improve the speed and efficiency of processing.

Type. In some instances, the Type node caches the input data when reading values; the cache is used for downstream processing. The cache requires sufficient disk space to store the entire data set but speeds up processing.

Evaluation. The Evaluation node must sort the input data to compute tiles. The sort is repeated for each model evaluated because the scores and consequent record order are different in each case. The running time is $M \log(N)$, where M is the number of models and N is the number of records.

Related information

- [Performance: Modeling Nodes](#)
- [Performance: CLEM Expressions](#)

Performance: Modeling Nodes

Neural Net and Kohonen. Neural network training algorithms (including the Kohonen algorithm) make many passes over the training data. The data is stored in memory up to a limit, and the excess is spilled to disk. Accessing the training data from disk is expensive because the access method is random, which can lead to excessive disk activity. You can disable the use of disk storage for these algorithms, forcing all data to be stored in memory, by selecting the Optimize for speed option on the Model tab of the node's dialog box. Note that if the amount of memory required to store the data is greater than the working set of the server process, part of it will be paged to disk and performance will suffer accordingly.

When Optimize for memory is enabled, a percentage of physical RAM is allocated to the algorithm according to the value of the IBM® SPSS® Modeler Server configuration option *Modeling memory limit percentage*. To use more memory for training neural networks, either provide more RAM or increase the value of this option, but note that setting the value too high will cause paging.

The running time of the neural network algorithms depends on the required level of accuracy. You can control the running time by setting a stopping condition in the node's dialog box.

K-Means. The K-Means clustering algorithm has the same options for controlling memory usage as the neural network algorithms. Performance on data stored on disk is better, however, because access to the data is sequential.

Related information

- [Performance: Process Nodes](#)
 - [Performance: CLEM Expressions](#)
-

Performance: CLEM Expressions

CLEM sequence functions (“@ functions”) that look back into the data stream must store enough of the data to satisfy the longest look-back. For operations whose degree of look-back is unbounded, all values of the field must be stored. An unbounded operation is one where the offset value is not a literal integer; for example, @OFFSET(Sales, Month). The offset value is the field name *Month*, whose value is unknown until executed. The server must save all values of the *Sales* field to ensure accurate results. Where an upper bound is known, you should provide it as an additional argument; for example, @OFFSET(Sales, Month, 12). This operation instructs the server to store no more than the 12 most recent values of *Sales*. Sequence functions, bounded or otherwise, almost always inhibit SQL generation.

Related information

- [Performance: Process Nodes](#)
 - [Performance: Modeling Nodes](#)
-

Accessibility in IBM® SPSS® Modeler

- [Overview of Accessibility in IBM SPSS Modeler](#)
 - [Types of Accessibility Support](#)
 - [Tips for use](#)
-

Overview of Accessibility in IBM SPSS Modeler

IBM® SPSS® Modeler provides accessibility support for all users, as well as specific support for users with visual and other functional impairments. This section describes the features and methods of working using accessibility enhancements, such as screen readers and keyboard shortcuts.

Related information

- [Types of Accessibility Support](#)
 - [Tips for use](#)
-

Types of Accessibility Support

Whether you have a visual impairment or are dependent on the keyboard for manipulation, there is a wide variety of alternative methods for using this data mining toolkit. For example, you can build streams, specify options, and read output, all without using the mouse. Available keyboard shortcuts are listed in the topics that follow. Additionally, IBM® SPSS® Modeler provides extensive support for screen readers, such as JAWS for Windows. You can also optimize the color scheme to provide additional contrast. These types of support are discussed in the following topics.

- [Accessibility for the Visually Impaired](#)
- [Accessibility for Blind Users](#)

- [Keyboard Accessibility](#)
- [Using a Screen Reader](#)

Related information

- [Overview of Accessibility in IBM SPSS Modeler](#)
- [Tips for use](#)
- [Accessibility for the Visually Impaired](#)
- [Accessibility for Blind Users](#)
- [Keyboard Accessibility](#)
- [Using a Screen Reader](#)
- [Using a Screen Reader with HTML Output](#)
- [Accessibility in the Interactive Tree Window](#)

Accessibility for the Visually Impaired

There are a number of properties you can specify in IBM® SPSS® Modeler that will enhance your ability to use the software.

Display Options

You can select colors for the display of graphs. You can also choose to use your specific Windows settings for the software itself. This may help to increase visual contrast.

1. To set display options, on the Tools menu, click User Options.
2. Click the Display tab. The options on this tab include the software color scheme, chart colors, and font sizes for nodes.

Note: The screen reader is not able to read graphs, so these are not accessible to visually-impaired users.

Use of Sounds for Notification

By turning sounds on or off, you can control the way you are alerted to particular operations in the software. For example, you can activate sounds for events such as node creation and deletion or the generation of new output or models.

1. To set notification options, on the Tools menu, click User Options.
2. Click the Notifications tab.

Controlling the Automatic Launching of New Windows

The Notifications tab on the User Options dialog box is also used to control whether newly generated output, such as tables and charts, are launched in a separate window. It may be easier for you to disable this option and open an output window only when required.

1. To set these options, on the Tools menu, click User Options.
2. Click the Notifications tab.
3. In the dialog box, select New Output from the list in the Visual Notifications group.
4. Under Open Window, select Never.

Node Size

Nodes can be displayed using either a standard or small size. You may want to adjust these sizes to fit your needs.

1. To set node size options, on the File menu, click Stream Properties.
2. Click the Layout tab.
3. From the Icon Size list, select Standard.

Related information

- [Types of Accessibility Support](#)
- [Accessibility for Blind Users](#)
- [Keyboard Accessibility](#)
- [Using a Screen Reader](#)
- [Using a Screen Reader with HTML Output](#)
- [Accessibility in the Interactive Tree Window](#)
- [Setting user options](#)

Accessibility for Blind Users

Support for blind users is predominately dependent on the use of a screen reader, such as JAWS for Windows. To optimize the use of a screen reader with IBM® SPSS® Modeler, you can specify a number of settings.

Display Options

Screen readers tend to perform better when the visual contrast is greater on the screen. If you already have a high-contrast Windows setting, you can choose to use these Windows settings for the software itself.

1. To set display options, on the Tools menu, click User Options.
2. Click the Display tab.

Note: The screen reader is not able to read graphs, so these are not accessible to blind users.

Use of Sounds for Notification

By turning on or off sounds, you can control the way you are alerted to particular operations in the software. For example, you can activate sounds for events such as node creation and deletion or the generation of new output or models.

1. To set notification options, on the Tools menu, click User Options.
2. Click the Notifications tab.

Controlling the Automatic Launching of New Windows

The Notifications tab on the User Options dialog box is also used to control whether newly generated output is launched in a separate window. It may be easier for you to disable this option and open an output window as needed.

1. To set these options, on the Tools menu, click User Options.
2. Click the Notifications tab.
3. In the dialog box, select New Output from the list in the Visual Notifications group.
4. Under Open Window, select Never.

Related information

- [Types of Accessibility Support](#)
- [Accessibility for the Visually Impaired](#)
- [Keyboard Accessibility](#)
- [Using a Screen Reader](#)
- [Using a Screen Reader with HTML Output](#)
- [Accessibility in the Interactive Tree Window](#)
- [Setting user options](#)

Keyboard Accessibility

The product's functionality is accessible from the keyboard. At the most basic level, you can press Alt plus the appropriate key to activate window menus (such as Alt+F to access the File menu) or press the Tab key to scroll through dialog box controls. However, there are special issues related to each of the product's main windows and helpful hints for navigating dialog boxes.

This section will cover the highlights of keyboard accessibility, from opening a stream to using node dialog boxes to working with output. Additionally, lists of keyboard shortcuts are provided for even more efficient navigation.

- [Shortcuts for navigating the main window](#)
- [Shortcuts for Dialog Boxes and Tables](#)
- [Shortcuts for Comments](#)
- [Shortcuts for Cluster Viewer and Model Viewer](#)
- [Shortcut Keys Example: Building Streams](#)
- [Shortcut Keys Example: Editing Nodes](#)

Related information

- [Types of Accessibility Support](#)

- [Accessibility for the Visually Impaired](#)
- [Accessibility for Blind Users](#)
- [Using a Screen Reader](#)
- [Using a Screen Reader with HTML Output](#)
- [Accessibility in the Interactive Tree Window](#)
- [Using shortcut keys](#)
- [Shortcuts for navigating the main window](#)
- [Shortcuts for Dialog Boxes and Tables](#)
- [Shortcuts for Comments](#)
- [Shortcuts for Cluster Viewer and Model Viewer](#)
- [Shortcut Keys Example: Building Streams](#)
- [Shortcut Keys Example: Editing Nodes](#)

Shortcuts for navigating the main window

You do most of your data mining work in the main window of IBM® SPSS® Modeler. The main area is called the **stream canvas** and is used to build and run data streams. The bottom part of the window contains the **node palettes**, which contain all available nodes. The palettes are organized on tabs corresponding to the type of data mining operation for each group of nodes. For example, nodes used to bring data into IBM SPSS Modeler are grouped on the Sources tab, and nodes used to derive, filter, or type fields are grouped on the Field Ops tab (short for Field Operations).

The right side of the window contains several tools for managing streams, output, and projects. The top half on the right contains the **managers** and has three tabs that are used to manage streams, output, and generated models. You can access these objects by selecting the tab and an object from the list. The bottom half on the right contains the **project pane**, which allows you to organize your work into projects. There are two tabs in this area reflecting two different views of a project. The **Classes view** sorts project objects by type, while the **CRISP-DM view** sorts objects by the relevant data mining phase, such as Data Preparation or Modeling. These various aspects of the IBM SPSS Modeler window are discussed throughout the Help system and User's Guide.

Following is a table of shortcuts used to move within the main IBM SPSS Modeler window and build streams. Shortcuts for dialog boxes and output are listed in the topics that follow. Note that these shortcut keys are available only from the main window.

Table 1. Main Window Shortcuts

Shortcut Key	Function
Ctrl+F5	Moves focus to the node palettes.
Ctrl+F6	Moves focus to the stream canvas.
Ctrl+F7	Moves focus to the managers pane.
Ctrl+F8	Moves focus to the project pane.

Table 2. Node and Stream Shortcuts

Shortcut Key	Function
Ctrl+N	Creates a new blank stream canvas.
Ctrl+O	Displays the Open dialog box, from where you can select and open an existing stream.
Ctrl+number keys	Moves focus to the corresponding tab on a window or pane. For example, within a tabbed pane or window, Ctrl+1 moves to the first tab starting from the left, Ctrl+2 to the second, etc.
Ctrl+Down Arrow	Used in the node palette to move focus from a palette tab to the first node under that tab.
Ctrl+Up Arrow	Used in the node palette to move focus from a node to its palette tab.
Enter	When a node is selected in the node palette (including refined models in the generated models palette), this keystroke adds the node to the stream canvas. Pressing Enter when a node is already selected on the canvas opens the dialog box for that node.
Ctrl+Enter	When a node is selected in the palette, adds that node to the stream canvas without selecting it, and moves focus to the first node in the palette.
Alt+Enter	When a node is selected in the palette, adds that node to the stream canvas and selects it, while moving focus to the first node in the palette.
Shift+Spacebar	When a node or comment has focus in the palette, toggles between selecting and deselecting that node or comment. If any other nodes or comments are also selected, this causes them to be deselected.
Ctrl+Shift+Spacebar	When a node or comment has focus in the stream, or a node or comment has focus on the palette, toggles between selecting and deselecting the node or comment. This does not affect any other selected nodes or comments.
Left/Right Arrow	If the stream canvas has focus, moves the entire stream horizontally on the screen. If a palette tab has focus, cycles between tabs. If a palette node has focus, moves between nodes in the palette.
Up/Down Arrow	If the stream canvas has focus, moves the entire stream vertically on the screen. If a palette node has focus, moves between nodes in the palette. If a subpalette has focus, moves between other subpalettes for this palette tab.

Shortcut Key	Function
Alt+Left/Right Arrow	Moves selected nodes and comments on the stream canvas horizontally in the direction of the arrow key.
Alt+Up/Down Arrow	Moves selected nodes and comments on the stream canvas vertically in the direction of the arrow key.
Ctrl+A	Selects all nodes in a stream.
Ctrl+Q	When a node has focus, selects it and all nodes downstream, and deselects all nodes upstream.
Ctrl+W	When a selected node has focus, deselects it and all selected nodes downstream.
Ctrl+Alt+D	Duplicates a selected node.
Ctrl+Alt+L	When a model nugget is selected in the stream, opens an Insert dialog box to enable you to load a saved model from a .nod file into the stream.
Ctrl+Alt+R	Displays the Annotations tab for a selected node, enabling you to rename the node.
Ctrl+Alt+U	Creates a User Input source node.
Ctrl+Alt+C	Toggles the cache for a node on or off.
Ctrl+Alt+F	Flushes the cache for a node.
Tab	On the stream canvas, cycles through all the source nodes and comments in the current stream. On a node palette, moves between nodes in the palette. On a selected subpalette, moves to the first node in the subpalette.
Shift+Tab	Performs the same operation as Tab but in reverse order.
Ctrl+Tab	With focus on the managers pane or project pane, moves focus to the stream canvas. With focus on a node palette, moves focus between a node and its palette tab.
Any alphabetic key	With focus on a node in the current stream, gives focus and cycles to the next node whose name starts with the key pressed.
F1	Opens the Help system at a topic relevant to the focus.
F2	Starts the connection process for a node selected in the canvas. Use the Tab key to move to the required node on the canvas, and press Shift+Spacebar to finish the connection.
F3	Deletes all connections for the selected node on the canvas.
F6	Moves focus between the managers pane, project pane and node palettes.
F10	Opens the File menu.
Shift+F10	Opens the pop-up menu for the node or stream.
Delete	Deletes a selected node from the canvas.
Esc	Closes a pop-up menu or dialog box.
Ctrl+Alt+X	Expands a SuperNode.
Ctrl+Alt+Z	Zooms in on a SuperNode.
Ctrl+Alt+Shift+Z	Zooms out of a SuperNode.
Ctrl+E	With focus in the stream canvas, this runs the current stream.

A number of standard shortcut keys are also used in IBM SPSS Modeler, such as Ctrl+C to copy. See the topic [Using shortcut keys](#) for more information.

Related information

- [Using shortcut keys](#)
- [Keyboard Accessibility](#)
- [Shortcuts for Dialog Boxes and Tables](#)
- [Shortcuts for Comments](#)
- [Shortcuts for Cluster Viewer and Model Viewer](#)
- [Shortcut Keys Example: Building Streams](#)
- [Shortcut Keys Example: Editing Nodes](#)
- [Using a Screen Reader](#)

Shortcuts for Dialog Boxes and Tables

Several shortcut and screen reader keys are helpful when you are working with dialog boxes, tables, and tables in dialog boxes. A complete list of special keyboard and screen reader shortcuts follows.

Table 1. Dialog Box and Expression Builder Shortcuts

Shortcut Key	Function
Alt+4	Used to dismiss all open dialog boxes or output windows. Output can be retrieved from the Outputs tab in the managers pane.
Ctrl+End	With focus on any control in the Expression Builder, this will move the insertion point to the end of the expression.

Shortcut Key	Function
Ctrl+1	In the Expression Builder, moves focus to the expression edit control.
Ctrl+2	In the Expression Builder, moves focus to the function list.
Ctrl+3	In the Expression Builder, moves focus to the field list.

Table Shortcuts

Table shortcuts are used for output tables as well as table controls in dialog boxes for nodes such as Type, Filter, and Merge. Typically, you will use the Tab key to move between table cells and Ctrl+Tab to leave the table control. *Note:* Occasionally, a screen reader may not immediately begin reading the contents of a cell. Pressing the arrow keys once or twice will reset the software and start the speech.

Table 2. Table Shortcuts

Shortcut Key	Function
Ctrl+W	For tables, reads the short description of the selected roW. For example, "Selected row 2 values are sex, flag, m/f, etc."
Ctrl+Alt+W	For tables, reads the long description of the selected roW. For example, "Selected row 2 values are field = sex, type = flag, sex = m/f, etc."
Ctrl+D	For tables, reads the short Description of the selected area. For example, "Selection is one row by six columns."
Ctrl+Alt+D	For tables, provides the long Description of the selected area. For example, "Selection is one row by six columns. Selected columns are Field, Type, Missing. Selected row is 1."
Ctrl+T	For tables, provides a short description of the selected columns. For example, "Fields, Type, Missing."
Ctrl+Alt+T	For tables, provides a long description of the selected columns. For example, "Selected columns are Fields, Type, Missing."
Ctrl+R	For tables, provides the number of Records in the table.
Ctrl+Alt+R	For tables, provides the number of Records in the table as well as column names.
Ctrl+I	For tables, reads the cell Information, or contents, for the cell that has focus.
Ctrl+Alt+I	For tables, reads the long description of cell Information (column name and contents of the cell) for the cell that has focus.
Ctrl+G	For tables, provides short General selection information.
Ctrl+Alt+G	For tables, provides long General selection information.
Ctrl+Q	For tables, provides a Quick toggle of the table cells. Ctrl+Q reads long descriptions, such as "Sex=Female," as you move through the table using the arrow keys. Selecting Ctrl+Q again will toggle to short descriptions (cell contents).
F8	For tables, when the focus is the table, sets the focus to the column header.
Spacebar	For tables, when the focus is the column header, enables column sorting.

Related information

- [Keyboard Accessibility](#)
- [Shortcuts for navigating the main window](#)
- [Shortcuts for Comments](#)
- [Shortcuts for Cluster Viewer and Model Viewer](#)
- [Shortcut Keys Example: Building Streams](#)
- [Shortcut Keys Example: Editing Nodes](#)
- [Using a Screen Reader](#)
- [Using shortcut keys](#)

Shortcuts for Comments

When working with on-screen comments, you can use the following shortcuts.

Table 1. Comment Shortcuts

Shortcut Key	Function
Alt+C	Toggles the show/hide comment feature.
Alt+M	Inserts a new comment if comments are currently displayed; shows comments if they are currently hidden.
Tab	On the stream canvas, cycles through all the source nodes and comments in the current stream.
Enter	When a comment has focus, indicates the start of editing.
Alt+Enter or Ctrl+Tab	Ends editing and saves editing changes.
Esc	Cancels editing. Changes made during editing are lost.
Alt+Shift+Up Arrow	Reduces the height of the text area by one grid cell (or one pixel) if snap-to-grid is on (or off).
Alt+Shift+Down Arrow	Increases the height of the text area by one grid cell (or one pixel) if snap-to-grid is on (or off).
Alt+Shift+Left Arrow	Reduces the width of the text area by one grid cell (or one pixel) if snap-to-grid is on (or off).
Alt+Shift+Right Arrow	Increases the width of the text area by one grid cell (or one pixel) if snap-to-grid is on (or off).

Related information

- [Keyboard Accessibility](#)
- [Shortcuts for navigating the main window](#)
- [Shortcuts for Dialog Boxes and Tables](#)
- [Shortcuts for Cluster Viewer and Model Viewer](#)
- [Shortcut Keys Example: Building Streams](#)
- [Shortcut Keys Example: Editing Nodes](#)
- [Using a Screen Reader](#)
- [Using shortcut keys](#)

Shortcuts for Cluster Viewer and Model Viewer

Shortcut keys are available for navigating around the Cluster Viewer and Model Viewer windows.

Table 1. General Shortcuts - Cluster Viewer and Model Viewer

Shortcut Key	Function
Tab	Moves focus to the next screen control.
Shift+Tab	Moves focus to the previous screen control.
Down Arrow	If a drop-down list has focus, opens the list or moves to the next item on the list. If a menu has focus, moves to the next item on the menu. If a thumbnail graph has focus, moves to the next one in the set (or to the first one if the last thumbnail has focus).
Up Arrow	If a drop-down list is open, moves to the previous item on the list. If a menu has focus, moves to the previous item on the menu. If a thumbnail graph has focus, moves to the previous one in the set (or to the last one if the first thumbnail has focus).
Enter	Closes an open drop-down list, or makes a selection on an open menu.
F6	Toggles focus between the left- and right-hand panes of the window.
Left and Right Arrows	If a tab has focus, moves to the previous or next tab. If a menu has focus, moves to the previous or next menu.
Alt+letter	Selects the button or menu having this letter underlined in its name.
Esc	Closes an open menu or drop-down list.

Cluster Viewer only

The Cluster Viewer has a Clusters view that contains a cluster-by-features grid.

To choose the Clusters view instead of the Model Summary view:

1. Press Tab repeatedly until the View button is selected.
2. Press Down Arrow twice to select Clusters.
From here you can select an individual cell within the grid:
3. Press Tab repeatedly until you arrive at the last icon in the visualization toolbar.

Figure 1. Show Visualization Tree icon



4. Press Tab once more, then Spacebar, then an arrow key.

The following keyboard shortcuts are now available:

Table 2. Cluster Viewer Shortcuts

Shortcut Key	Function
Arrow key	Moves focus between individual cells in the grid. The cell distribution display in the right-hand pane changes as the focus moves.
Ctrl+, (comma)	Selects or deselects the entire column in the grid in which a cell has focus. To add a column to the selection, use the arrow keys to navigate to a cell in that column and press Ctrl+, again.
Tab	Moves focus out of the grid and onto the next screen control.
Shift+Tab	Moves focus out of the grid and back to the previous screen control.
F2	Enters edit mode (label and description cells only).
Enter	Saves editing changes and exits edit mode (label and description cells only).
Esc	Exits edit mode without saving changes (label and description cells only).

Related information

- [Keyboard Accessibility](#)
 - [Shortcuts for navigating the main window](#)
 - [Shortcuts for Dialog Boxes and Tables](#)
 - [Shortcuts for Comments](#)
 - [Shortcut Keys Example: Building Streams](#)
 - [Shortcut Keys Example: Editing Nodes](#)
 - [Using a Screen Reader](#)
 - [Using shortcut keys](#)
-

Shortcut Keys Example: Building Streams

To make the stream-building process more clear for users dependent on the keyboard or on a screen reader, following is an example of building a stream without the use of the mouse. In this example, you will build a stream containing a Variable File node, a Derive node, and a Histogram node using the following steps:

1. **Start IBM® SPSS® Modeler.** When IBM SPSS Modeler first starts, focus is on the Favorites tab of the node palette.
2. **Ctrl+Down Arrow.** Moves focus from the tab itself to the body of the tab.
3. **Right Arrow.** Moves focus to the Variable File node.
4. **Spacebar.** Selects the Variable File node.
5. **Ctrl+Enter.** Adds the Variable File node to the stream canvas. This key combination also keeps selection on the Variable File node so that the next node added will be connected to it.
6. **Tab.** Moves focus back to the node palette.
7. **Right Arrow 4 times.** Moves to the Derive node.
8. **Spacebar.** Selects the Derive node.
9. **Alt+Enter.** Adds the Derive node to the canvas and moves selection to the Derive node. This node is now ready to be connected to the next added node.
10. **Tab.** Moves focus back to the node palette.
11. **Right Arrow 5 times.** Moves focus to the Histogram node in the palette.
12. **Spacebar.** Selects the Histogram node.
13. **Enter.** Adds the node to the stream and moves focus to the stream canvas.

Continue with the next example, or save the stream if you want to try the next example at a later time.

Related information

- [Using shortcut keys](#)
 - [Keyboard Accessibility](#)
 - [Shortcuts for navigating the main window](#)
 - [Shortcuts for Dialog Boxes and Tables](#)
 - [Shortcuts for Comments](#)
 - [Shortcuts for Cluster Viewer and Model Viewer](#)
 - [Shortcut Keys Example: Editing Nodes](#)
 - [Using a Screen Reader](#)
-

Shortcut Keys Example: Editing Nodes

In this example, you will use the stream built in the earlier example. The stream consists of a Variable File node, a Derive node, and a Histogram node. The instructions begin with focus on the third node in the stream, the Histogram node.

1. **Ctrl+Left Arrow 2 times.** Moves focus back to the Variable File node.
2. **Enter.** Opens the Variable File dialog box. Tab through to the File field and type a text file path and name to select that file. Press Ctrl+Tab to navigate to the lower part of the dialog box, tab through to the OK button and press Enter to close the dialog box.
3. **Ctrl+Right Arrow.** Gives focus to the second node, a Derive node.
4. **Enter.** Opens the Derive node dialog box. Tab through to select fields and specify derive conditions. Press Ctrl+Tab to navigate to the OK button and press Enter to close the dialog box.
5. **Ctrl+Right Arrow.** Gives focus to the third node, a Histogram node.
6. **Enter.** Opens the Histogram node dialog box. Tab through to select fields and specify graph options. For drop-down lists, press Down Arrow to open the list and to highlight a list item, then press Enter to select the list item. Tab through to the OK button and press Enter to close the dialog box.

At this point, you can add additional nodes or run the current stream. Keep in mind the following tips when you are building streams:

- When manually connecting nodes, use F2 to create the start point of a connection, tab to move to the end point, then use Shift+Spacebar to finalize the connection.

- Use F3 to destroy all connections for a selected node in the canvas.
- Once you have created a stream, use Ctrl+E to run the current stream.

A complete list of shortcut keys is available. See the topic [Shortcuts for navigating the main window](#) for more information.

Related information

- [Using shortcut keys](#)
 - [Keyboard Accessibility](#)
 - [Shortcuts for navigating the main window](#)
 - [Shortcuts for Dialog Boxes and Tables](#)
 - [Shortcuts for Comments](#)
 - [Shortcuts for Cluster Viewer and Model Viewer](#)
 - [Shortcut Keys Example: Building Streams](#)
 - [Using a Screen Reader](#)
-

Using a Screen Reader

A number of screen readers are available on the market. IBM® SPSS® Modeler is configured to support JAWS for Windows using the Java Access Bridge, which is installed along with IBM SPSS Modeler. If you have JAWS installed, simply launch JAWS before launching IBM SPSS Modeler to use this product.

Note: We recommend that you have at least 6GB space to run JAWS with SPSS Modeler.

Due to the nature of IBM SPSS Modeler's unique graphical representation of the data mining process, charts and graphs are optimally used visually. It is possible, however, for you to understand and make decisions based on output and models viewed textually using a screen reader.

Note: With 64-bit client machines, some assistive technology features do not work. This is because the Java Access Bridge is not designed for 64-bit operation.

Using the IBM SPSS Modeler Dictionary File

An IBM SPSS Modeler dictionary file (*Awt.JDF*) is available for inclusion with JAWS. To use this file:

1. Navigate to the */accessibility* subdirectory of your IBM SPSS Modeler installation and copy the dictionary file (*Awt.JDF*).
2. Copy it to the directory with your JAWS scripts.

You may already have a file named *Awt.JDF* on your machine if you have other JAVA applications running. In this case, you may not be able to use this dictionary file without manually editing the dictionary file.

- [Using a Screen Reader with HTML Output](#)
- [Accessibility in the Interactive Tree Window](#)

Related information

- [Types of Accessibility Support](#)
 - [Accessibility for the Visually Impaired](#)
 - [Accessibility for Blind Users](#)
 - [Keyboard Accessibility](#)
 - [Using a Screen Reader with HTML Output](#)
 - [Accessibility in the Interactive Tree Window](#)
 - [Shortcuts for navigating the main window](#)
 - [Shortcuts for Dialog Boxes and Tables](#)
 - [Shortcuts for Comments](#)
 - [Shortcuts for Cluster Viewer and Model Viewer](#)
 - [Shortcut Keys Example: Building Streams](#)
 - [Shortcut Keys Example: Editing Nodes](#)
-

Using a Screen Reader with HTML Output

When viewing output displayed as HTML within IBM® SPSS® Modeler using a screen reader, you may encounter some difficulties. A number of types of output are affected, including:

- Output viewed on the Advanced tab for Regression, Logistic Regression, and Factor/PCA nodes
- Report node output

In each of these windows or dialog boxes, there is a tool on the toolbar that can be used to launch the output into your default browser, which provides standard screen reader support. You can then use the screen reader to convey the output information.

Related information

- [Types of Accessibility Support](#)
 - [Accessibility for the Visually Impaired](#)
 - [Accessibility for Blind Users](#)
 - [Keyboard Accessibility](#)
 - [Using a Screen Reader](#)
 - [Accessibility in the Interactive Tree Window](#)
-

Accessibility in the Interactive Tree Window

The standard display of a decision tree model in the Interactive Tree window may cause problems for screen readers. To access an accessible version, on the Interactive Tree menus click:

View > Accessible Window

This displays a view similar to the standard tree map, but one which JAWS can read correctly. You can move up, down, right, or left using the standard arrow keys. As you navigate the accessible window, the focus in the Interactive Tree window moves accordingly. Use the Spacebar to change the selection, or use Ctrl+Spacebar to extend the current selection.

Related information

- [Types of Accessibility Support](#)
 - [Accessibility for the Visually Impaired](#)
 - [Accessibility for Blind Users](#)
 - [Keyboard Accessibility](#)
 - [Using a Screen Reader](#)
 - [Using a Screen Reader with HTML Output](#)
-

Tips for use

There are several tips for making the IBM® SPSS® Modeler environment more accessible to you. The following are general hints when working in IBM SPSS Modeler.

- Exiting extended text boxes. Use Ctrl+Tab to exit extended text boxes. Note that Ctrl+Tab is also used to exit table controls.
- Using the Tab key rather than arrow keys. When selecting options for a dialog box, use the Tab key to move between option buttons. The arrow keys will not work in this context.
- Drop-down lists. In a drop-down list for dialog boxes, you can use either the Escape key or the space bar to select an item and then close the list. You can also use the Escape key to close drop-down lists that do not close when you have tabbed to another control.
- Execution status. When you are running a stream on a large database, JAWS can lag behind in reading the stream status to you. Press the Ctrl key periodically to update the status reporting.
- Using the node palettes. When you first enter a tab of the node palettes, JAWS will sometimes read "groupbox" instead of the name of the node. In this case, you can use Ctrl+Right Arrow and then Ctrl+Left Arrow to reset the screen reader and hear the node name.
- Reading menus. Occasionally, when you are first opening a menu, JAWS may not read the first menu item. If you suspect that this may have happened, use the Down Arrow and then the Up Arrow to hear the first item in the menu.
- Cascaded menus. JAWS does not read the first level of a cascaded menu. If you hear a break in speaking while moving through a menu, press the Right Arrow key to hear the child menu items.

Additionally, if you have IBM SPSS Modeler Text Analytics installed, the following tips can make the interactive workbench interface more accessible to you.

- Entering dialog boxes. You may need to press the Tab key to put the focus on the first control upon entering a dialog box.
- Exiting extended text boxes. Use Ctrl+Tab to exit extended text boxes and move to the next control. Note that Ctrl+Tab is also used to exit table controls.
- Typing the first letter to find element in tree list. When looking for an element in the categories pane, extracted results pane, or library tree, you can type the first letter of the element when the pane has the focus. This will select the next occurrence of an element beginning with the letter you entered.
- Drop-down lists. In a drop-down list for dialog boxes, you can use the space bar to select an item and then close the list.

Additional tips for use are discussed at length in the following topics.

- [Interference with Other Software](#)
 - [JAWS and Java](#)
 - [Using Graphs in IBM SPSS Modeler](#)
-

Interference with Other Software

When testing IBM® SPSS® Modeler with screen readers, such as JAWS, our development team discovered that the use of a Systems Management Server (SMS) within your organization may interfere with JAWS' ability to read Java-based applications, such as IBM SPSS Modeler. Disabling SMS will correct this situation. Visit the Microsoft website for more information on SMS.

Related information

- [Tips for use](#)
 - [JAWS and Java](#)
 - [Using Graphs in IBM SPSS Modeler](#)
-

JAWS and Java

Different versions of JAWS provide varying levels of support for Java-based software applications. Although IBM® SPSS® Modeler will work with all recent versions of JAWS, some versions may have minor problems when used with Java-based systems. Visit the JAWS for Windows website at <http://www.FreedomScientific.com>.

Related information

- [Tips for use](#)
 - [Interference with Other Software](#)
 - [Using Graphs in IBM SPSS Modeler](#)
-

Using Graphs in IBM® SPSS® Modeler

Visual displays of information, such as histograms, evaluation charts, multiplots, and scatterplots, are difficult to interpret with a screen reader. Please note, however, that web graphs and distributions can be viewed using the textual summary available from the output window.

Related information

- [Tips for use](#)
 - [Interference with Other Software](#)
 - [JAWS and Java](#)
-

Unicode support

- [Unicode Support in IBM SPSS Modeler](#)
-

Unicode Support in IBM SPSS Modeler

IBM® SPSS® Modeler is fully Unicode-enabled for both IBM SPSS Modeler and IBM SPSS Modeler Server. This makes it possible to exchange data with other applications that support Unicode, including multi-language databases, without any loss of information that might be caused by conversion to or from a locale-specific encoding scheme.

- IBM SPSS Modeler stores Unicode data internally and can read and write multi-language data stored as Unicode in databases without loss.
- IBM SPSS Modeler can read and write UTF-8 encoded text files. Text file import and export will default to the locale-encoding but support UTF-8 as an alternative. This setting can be specified in the file import and export nodes, or the default encoding can be changed in the stream properties dialog box. See the topic [Setting general options for streams](#) for more information.

- Statistics, SAS, and Text data files stored in the locale-encoding will be converted to UTF-8 on import and back again on export. When writing to any file, if there are Unicode characters that do not exist in the locale character set, they will be substituted and a warning will be displayed. This should occur only where the data has been imported from a data source that supports Unicode (a database or UTF-8 text file) and that contains characters from a different locale or from multiple locales or character sets.
- IBM SPSS Modeler Solution Publisher images are UTF-8 encoded and are truly portable between platforms and locales.

About Unicode

The goal of the Unicode standard is to provide a consistent way to encode multilingual text so that it can be easily shared across borders, locales, and applications. The Unicode Standard, now at version 4.0.1, defines a character set that is a superset of all of the character sets in common use in the world today and assigns to each character a unique name and code point. The characters and their code points are identical to those of the Universal Character Set (UCS) defined by ISO-10646. For more information, see the [Unicode Home Page](#).

Batch Mode Execution

- [Introduction to Batch Mode](#)
- [Working in Batch Mode](#)

Introduction to Batch Mode

Data mining is usually an interactive process—you interact with data and models to improve your understanding of the data and the domain it represents. However, IBM® SPSS® Modeler streams can also be used to process data and perform data mining tasks in **batch mode**, with no visible user interface. Batch mode allows long-running or repetitive tasks to be performed without your intervention and without the presence of the user interface on the screen.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > About > Additional Details

If connectivity is enabled, you see the option Server Enablement in the License Status tab.

See [Connecting to IBM SPSS Modeler Server](#) for more information.

Examples of tasks appropriate for batch mode include:

- Running a time-consuming modeling exercise in the background.
- Running a stream at a scheduled time (for example, overnight, when the resultant load on the computer will not be inconvenient).
- Running a data preprocessing stream on a large volume of data (for example, in the background and/or overnight).
- Running regularly scheduled tasks, such as monthly reports.
- Running a stream as an embedded part of another process, such as a scoring engine facility.

Note: IBM SPSS Modeler operations can be scheduled in batch mode using the appropriate operating system commands or utilities (for example, the `at` command under Windows NT).

Related information

- [Working in Batch Mode](#)
- [Invoking the Software](#)
- [Batch mode log files](#)
- [Scripting in Batch Mode](#)
- [Using Parameters in Batch Mode](#)
- [Working with Output in Batch Mode](#)

Working in Batch Mode

Working in batch mode typically involves:

1. Invoking IBM® SPSS® Modeler in batch mode using the `clemb` command.
2. Connecting to a server.
3. Loading an existing stream or script file.
4. Executing the stream or script.

Note: SPSS Modeler Batch requires 4GB of available memory.

Once execution is complete, you can then consult the log file produced by default in batch mode and view the results of graphs, output nodes, and models. For more information about these steps, see the following topics.

If only SPSS Modeler Batch is installed (not SPSS Modeler client), to run a stream containing a Statistics node, you also need to complete the following steps:

1. Install SPSS Modeler Server and IBM SPSS Statistics Server on the same machine.
2. Run a utility on the SPSS Modeler Server host machine to create the statistics.ini file, which tells IBM SPSS Statistics the installation path for SPSS Modeler Server. To run the utility, open a command prompt, change to the SPSS Modeler Server bin directory, and run the following command.

On Windows:

```
statisticsutility -location=<statistics_installation_path>/bin
```

On Linux:

```
./statisticsutility -location=<statistics_installation_path>/bin
```

3. Run the batch command. For example:

```
clemb -server -hostname 9.30.51.42 -port 28181 -username xxxxxxxxx -password xxxxxxxxx -stream "c:\test\StatisticsOutputNode.str" -execute -log "c:\log\report.log"
```

- [Invoking the Software](#)
- [Using Command Line Arguments](#)
- [Batch mode log files](#)
- [Scripting in Batch Mode](#)
- [Using Parameters in Batch Mode](#)
- [Working with Output in Batch Mode](#)

Invoking the Software

You can use the command line of your operating system to launch IBM® SPSS® Modeler as follows:

1. On a computer where IBM SPSS Modeler is installed, open a DOS, or command-prompt, window.
2. To launch the IBM SPSS Modeler interface in interactive mode, type the **modelerclient** command followed by the required arguments; for example:

```
modelerclient -stream report.str -execute
```

The available arguments (flags) allow you to connect to a server, load streams, run scripts, or specify other parameters as needed.

Related information

- [Introduction to Batch Mode](#)
- [Batch mode log files](#)
- [Scripting in Batch Mode](#)
- [Using Parameters in Batch Mode](#)
- [Working with Output in Batch Mode](#)

Using Command Line Arguments

In order for IBM® SPSS® Modeler to open and execute files (such as streams and scripts) in batch mode, you need to alter the initial command (**clemb**) that launches the software. There are a number of command line arguments, also referred to as **flags**, that you can use to:

- Connect to a server.
- Load streams, scripts, models, states, projects, and output files. (If you have licensed IBM SPSS Collaboration and Deployment Services Repository, you can connect to a repository and load objects from it.)
- Specify log file options.
- Set default directories for use in IBM SPSS Modeler.

All of the above operations require the use of flags appended to the **clemb** command. Flags follow the form **-flag**, where the hyphen precedes the argument itself. For example, using the flag **-server** in conjunction with the initial argument **clemb** will connect to the server specified using other flag options.

You can combine the `clem` command with a number of other startup flags, such as `-server`, `-stream`, and `-execute`, in order to load and execute streams in batch mode. The following command loads and executes the stream `report.str` without invoking the user interface:

```
clem -server -hostname myserver -port 80  
-username dminer -password 1234 -stream report.str -execute
```

For a complete list of command line arguments, see [Command Line Arguments](#).

- IBM SPSS Modeler states and scripts are also executed in this manner, using the `-state` and `-script` flags, respectively. Multiple states and streams can be loaded by specifying the relevant flag for each item.
- Multiple arguments can be combined into a single command file and specified at startup using the @ symbol. See the topic [Combining Multiple Arguments](#) for more information.

Related information

- [Invoking the Software](#)
- [Invoking the software](#)
- [Using command line arguments](#)
- [System arguments](#)
- [Server connection arguments](#)
- [IBM SPSS Collaboration and Deployment Services Repository Connection Arguments](#)
- [Introduction to Batch Mode](#)

Batch mode log files

Running in batch mode produces a log file. By default, the name of this log file is `clem_batch.log`, but you can specify an alternative name using the `-log` flag. For example, the following command runs `report.str` in batch mode and sends the logging information to `report.log`:

```
clem -server -hostname myserver -port 80  
-username dminer -password 1234 -stream report.str  
-execute -log report.log
```

Normally, the log file overwrites any existing file of the same name, but you can make IBM® SPSS® Modeler append to the log file instead by using the `-appendlog` flag. Logging can also be suppressed altogether by using the `-nolog` flag.

Note: Logging arguments are available only when running in batch mode.

On **Windows** and **Linux**, by default, `clem_batch.log` is generated to the same location as the `clem` command (for example, <installation_path>/bin on Windows and <installation_path>/ on Linux).

On **Mac OS**, by default, `clem_batch.log` is generated to <installation_path>/IBM SPSS Modeler.app/Contents/log.

Scripting in Batch Mode

In its simplest form, batch mode execution of IBM® SPSS® Modeler streams is performed one at a time using the command line arguments discussed in this guide. A given stream is executed without significantly altering its node parameters. While this may work well for automated production of monthly churn reports or predictions, it cannot handle the sophisticated processes that many advanced data miners would like to automate.

For example, a financial institution may want to construct a number of models using different data or modeling parameters, test the models on another set of data, and produce a report on the results. Because this process requires repetitive modifications to a stream and the creation and deletion of nodes, automating it requires the use of scripting. Scripting allows complex processes that would otherwise require user intervention to be automated and executed in batch mode.

To Execute a Script in Batch Mode

1. Append the `clem` command with the `-script` flag, specifying the name of the script that you want to execute.
2. Also use the `-execute` flag with the above arguments to execute the specified script. This will run the stand-alone script in its entirety.

For example, to load and execute a script that runs a model producing churn scores that are stored as output for the data warehouse, you would use the following command:

```
clem -server -hostname myserver -port 80  
-username dminer -password 1234  
-script clemscript.txt -execute
```

Related information

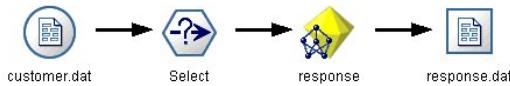
- [Introduction to Batch Mode](#)
- [Invoking the Software](#)
- [Batch mode log files](#)
- [Using Parameters in Batch Mode](#)
- [Working with Output in Batch Mode](#)

Using Parameters in Batch Mode

You can modify the effect of executing a stream in batch mode by supplying parameters to the command line launch of IBM® SPSS® Modeler. These might be **simple parameters** used directly in CLEM expressions, or they might be node properties, also called **slot parameters**, which are used to modify the settings of nodes in the stream.

For example, the following stream selects a subset of data from a file, passes it through a neural net, and sends the results to a file:

Figure 1. Stream operations in the user interface



The value of the field *Month* determines the selected data; the expression in the Select node is:

```
Month == '$P-mth'
```

When running the same stream in batch mode, select the appropriate month by setting the value of the parameter *mth* in the command line:

```
clemb -server -hostname myserver -port 80
-username dminer -password 1234
-stream predict.str -Pmth=Jan -execute
```

Note: In command line arguments, the **-P** flag is used to denote a parameter.

Sometimes the required command line control of the stream involves modifying the settings of the nodes in the stream using slot parameters. Consider the following stream, which reads a file, processes its contents, and sends a report to another file:

Figure 2. Stream operations in the user interface



Suppose that you want to generate the report once a month, reading the appropriate month's data and sending the report to a file whose name indicates the relevant month. You might want to set the filenames for the source data and for the report. The following command sets the appropriate slot parameters and executes the stream:

```
clemb -stream report.str -Porder.full_filename=APR_orders.dat
-Preport.filename=APR_report.txt -execute
```

Note: This command does not contain the operating-system-specific code that schedules it to run monthly.

Related information

- [Introduction to Batch Mode](#)
- [Invoking the Software](#)
- [Batch mode log files](#)
- [Scripting in Batch Mode](#)
- [Working with Output in Batch Mode](#)

Working with Output in Batch Mode

Working with visual output, such as tables, graphs, and charts, typically requires a user interface. Since batch mode does not launch the IBM® SPSS® Modeler user interface, output objects are diverted to a file so that you can view them later, either in the user interface or in another software package. Using the properties available for nodes (slot parameters), you can control the formats and filenames of output objects created during batch mode.

Related information

- [Introduction to Batch Mode](#)
 - [Invoking the Software](#)
 - [Batch mode log files](#)
 - [Scripting in Batch Mode](#)
 - [Using Parameters in Batch Mode](#)
-

Source Nodes

- [Overview](#)
- [Setting Field Storage and Formatting](#)
- [Unsupported control characters](#)
- [Analytic Server Source node](#)
- [Database source node](#)
- [Variable File Node](#)
- [Fixed File Node](#)
- [Statistics File Node](#)
- [Data Collection Node](#)
- [IBM Cognos source node](#)
- [IBM Cognos TM1 Source Node](#)
- [TWC source node](#)
- [SAS Source Node](#)
- [Excel Source node](#)
- [XML Source Node](#)
- [User Input Node](#)
- [Simulation Generate Node](#)
- [Extension Import node](#)
- [Geospatial Source Node](#)
- [JSON Source node](#)
- [Common Source Node Tabs](#)

Overview

Source nodes enable you to import data stored in a number of formats, such as flat files, IBM® SPSS® Statistics (.sav), SAS, Microsoft Excel, and ODBC-compliant relational databases. You can also generate synthetic data using the User Input node.

The Sources palette contains the following nodes:

	The Analytic Server source enables you to run a stream on Hadoop Distributed File System (HDFS). The information in an Analytic Server data source can come from a variety of places, such as text files and databases. See the topic Analytic Server Source node for more information.
	The Database node can be used to import data from a variety of other packages using ODBC (Open Database Connectivity), including Microsoft SQL Server, Db2, Oracle, and others. See the topic Database source node for more information.
	The Variable File node reads data from free-field text files—that is, files whose records contain a constant number of fields but a varied number of characters. This node is also useful for files with fixed-length header text and certain types of annotations. See the topic Variable File Node for more information.
	The Fixed File node imports data from fixed-field text files—that is, files whose fields are not delimited but start at the same position and are of a fixed length. Machine-generated or legacy data are frequently stored in fixed-field format. See the topic Fixed File Node for more information.
	The Statistics File node reads data from the .sav or .zsav file format used by IBM SPSS Statistics, as well as cache files saved in IBM SPSS Modeler, which also use the same format.
	The Data Collection node imports survey data from various formats used by market research software conforming to the Data Collection Data Model. A Data Collection Developer Library must be installed to use this node. See the topic Data Collection Node for more information.
	The IBM Cognos source node imports data from Cognos Analytics databases.
	The IBM Cognos TM1 source node imports data from Cognos TM1 databases.

	The SAS File node imports SAS data into IBM SPSS Modeler. See the topic SAS Source Node for more information.
	The Excel node imports data from Microsoft Excel in the .xlsx file format. An ODBC data source is not required. See the topic Excel Source node for more information.
	The XML source node imports data in XML format into the stream. You can import a single file, or all files in a directory. You can optionally specify a schema file from which to read the XML structure.
	The User Input node provides an easy way to create synthetic data—either from scratch or by altering existing data. This is useful, for example, when you want to create a test dataset for modeling. See the topic User Input Node for more information.
	The Simulation Generate node provides an easy way to generate simulated data—either from scratch using user specified statistical distributions or automatically using the distributions obtained from running a Simulation Fitting node on existing historical data. This is useful when you want to evaluate the outcome of a predictive model in the presence of uncertainty in the model inputs.
	Use the Geospatial source node to bring map or spatial data into your data mining session. See the topic Geospatial Source Node for more information.
	The JSON source node imports data from a JSON file. See JSON Source node for more information.

To begin a stream, add a source node to the stream canvas. Next, double-click the node to open its dialog box. The various tabs in the dialog box allow you to read in data; view the fields and values; and set a variety of options, including filters, data types, field role, and missing-value checking.

Setting Field Storage and Formatting

Options on the Data tab for Fixed File, Variable File, XML Source, and User Input nodes allow you to specify the storage type for fields as they are imported or created in IBM® SPSS® Modeler. For Fixed File, Variable File and User Input nodes you can also specify the field formatting, and other metadata.

For data read from other sources, storage is determined automatically but can be changed using a conversion function, such as `to_integer`, in a Filler node or Derive node.

Field Use the Field column to view and select fields in the current dataset.

Override Select the check box in the Override column to activate options in the Storage and Input Format columns.

Data Storage

Storage describes the way data are stored in a field. For example, a field with values of 1 and 0 stores integer data. This is distinct from the measurement level, which describes the usage of the data, and does not affect storage. For example, you may want to set the measurement level for an integer field with values of 1 and 0 to *Flag*. This usually indicates that 1 = *True* and 0 = *False*. While storage must be determined at the source, measurement level can be changed using a Type node at any point in the stream. See the topic [Measurement levels](#) for more information.

Available storage types are:

- String Used for fields that contain non-numeric data, also called alphanumeric data. A string can include any sequence of characters, such as *fred*, *Class 2*, or *1234*. Note that numbers in strings cannot be used in calculations.
- Integer A field whose values are integers.
- Real Values are numbers that may include decimals (not limited to integers). The display format is specified in the Stream Properties dialog box and can be overridden for individual fields in a Type node (Format tab).
- Date Date values specified in a standard format such as year, month, and day (for example, 2007–09–26). The specific format is specified in the Stream Properties dialog box.
- Time Time measured as a duration. For example, a service call lasting 1 hour, 26 minutes, and 38 seconds might be represented as 01:26:38, depending on the current time format as specified in the Stream Properties dialog box.
- Timestamp Values that include both a date and time component, for example 2007–09–26 09:04:00, again depending on the current date and time formats in the Stream Properties dialog box. Note that timestamp values may need to be wrapped in double-quotes to ensure they are interpreted as a single value rather than separate date and time values. (This applies for example when entering values in a User Input node.)
- List Introduced in SPSS Modeler version 17, along with new measurement levels of Geospatial and Collection, a List storage field contains multiple values for a single record. There are list versions of all of the other storage types.

Table 1. List storage type icons

Icon	Storage type
	List of string

Icon	Storage type
[#]	List of integer
[#]	List of real
[#]	List of time
[#]	List of date
[#]	List of timestamp
[#]	List with a depth greater than zero

In addition, for use with the Collection measurement level, there are list versions of the following measurement levels.

Table 2. List measurement level icons

Icon	Measurement level
[#]	List of continuous
[#]	List of categorical
[#]	List of flags
[#]	List of nominal
[#]	List of ordinal

Lists can be imported into SPSS Modeler in one of three source nodes (Analytic Server, Geospatial, or Variable File), or created within your streams through use of the Derive or Filler field operation nodes.

For more information on Lists and their interaction with the Collection and Geospatial measurement levels, see [List storage and associated measurement levels](#)

Storage conversions. You can convert storage for a field using a variety of conversion functions, such as `to_string` and `to_integer`, in a Filler node. See the topic [Storage Conversion Using the Filler Node](#) for more information. Note that conversion functions (and any other functions that require a specific type of input such as a date or time value) depend on the current formats specified in the Stream Properties dialog box. For example, if you want to convert a string field with values *Jan 2018, Feb 2018*, (and so forth) to date storage, select MON YYYY as the default date format for the stream. Conversion functions are also available from the Derive node, for temporary conversion during a derive calculation. You can also use the Derive node to perform other manipulations, such as recoding string fields with categorical values. See the topic [Recoding Values with the Derive Node](#) for more information.

Reading in mixed data. Note that when reading in fields with numeric storage (either integer, real, time, timestamp, or date), any non-numeric values are set to null or system missing. This is because unlike some applications, IBM SPSS Modeler does not allow mixed storage types within a field. To avoid this, any fields with mixed data should be read in as strings, either by changing the storage type in the source node or in the external application as necessary.

Field Input Format (Fixed File, Variable File, and User Input nodes only)

For all storage types except String and Integer, you can specify formatting options for the selected field using the drop-down list. For example, when merging data from various locales, you may need to specify a period (.) as the decimal separator for one field, while another will require a comma separator.

Input options specified in the source node override the formatting options specified in the stream properties dialog box; however, they do not persist later in the stream. They are intended to parse input correctly based on your knowledge of the data. The specified formats are used as a guide for parsing the data as they are read into IBM SPSS Modeler, not to determine how they should be formatted after being read into IBM SPSS Modeler. To specify formatting on a per-field basis elsewhere in the stream, use the Format tab of a Type node. See the topic [Field Format Settings Tab](#) for more information.

Options vary depending on the storage type. For example, for the Real storage type, you can select Period (.) or Comma (,) as the decimal separator. For timestamp fields, a separate dialog box opens when you select Specify from the drop-down list. See the topic [Setting Field Format options](#) for more information.

For all storage types, you can also select Stream default to use the stream default settings for import. Stream settings are specified in the stream properties dialog box.

Additional Options

Several other options can be specified using the Data tab:

- To view storage settings for data that are no longer connected through the current node (train data, for example), select View unused field settings. You can clear the legacy fields by clicking Clear.
- At any point while working in this dialog box, click Refresh to reload fields from the data source. This is useful when you are altering data connections to the source node or when you are working between tabs on the dialog box.
- [List storage and associated measurement levels](#)

Related information

- [Common Source Node Tabs](#)

List storage and associated measurement levels

Introduced in SPSS® Modeler version 17, to work with the new measurement levels of Geospatial and Collection, a List storage field contains multiple values for a single record. Lists are enclosed in square brackets ([]). Examples of lists are: [1,2,4,16] and ["abc", "def"].

Lists can be imported into SPSS Modeler in one of three source nodes (Analytic Server, Geospatial, or Variable File), created within your streams through use of the Derive or Filler field operation nodes, or generated by the Merge node when using the Ranked Condition merge method.

Lists are considered to have a depth; for example, a simple list with items that are contained within single square brackets, in the format [1,3], is recorded in IBM® SPSS Modeler with a depth of zero. In addition to simple lists that have a depth of zero, you can use nested lists, where each value within a list is a list itself.

The depth of a nested list depends on the associated measurement level. For Typeless there is no set depth limit, for Collection the depth is zero, and for Geospatial, the depth must be between zero and two inclusive depending on the number of nested items.

For zero depth lists you can set the measurement level as either Geospatial or Collection. Both of these levels are parent measurement levels and you set the measurement sublevel information in the Values dialog box. The measurement sublevel of a Collection determines the measurement level of the elements within that list. All measurement levels (except typeless and geospatial) are available as sublevels for Collections. The Geospatial measurement level has six sublevels of Point, LineString, Polygon, MultiPoint, MultiLineString, and MultiPolygon; for more information, see [Geospatial measurement sublevels](#).

Note: The Collection measurement level can only be used with lists of depth 0, the Geospatial measurement level can only be used with lists of a maximum depth of 2, and the Typeless measurement level can be used with any list depth.

The following example shows the difference between a zero depth list and a nested list by using the structure of the Geospatial measurement sublevels Point and LineString:

- The Geospatial measurement sublevel of Point has a field depth of zero:

[1,3] two coordinates

[1,3,-1] three coordinates

- The Geospatial measurement sublevel of LineString has a field depth of one:

[[1,3], [5,0]] two coordinates

[[1,3,-1], [5,0,8]] three coordinates

The Point field (with a depth of zero) is a normal list where each value is made up of two or three coordinates. The LineString field (with a depth of one) is a list of points, where each point is made up of a further series of list values.

For more information about list creation, see [Deriving a list or geospatial field](#).

Related information

- [Type Node](#)
- [Viewing and setting information about types](#)
- [Converting Continuous Data](#)
- [What is instantiation?](#)
- [Data Values](#)
- [Checking Type Values](#)
- [Setting the field role](#)

Unsupported control characters

Some of the processes in SPSS® Modeler cannot handle data that includes various control characters. If your data uses these characters you may see an error message such as the following example:

Unsupported control characters found in values of field {0}

The unsupported characters are: from 0x0 to 0x3F inclusive, and 0x7F; however, the tab (0x9(\t)), new line (0xA(\n)), and carriage return (0xD(\r)) characters do not cause a problem.

If you see an error message that relates to unsupported characters, in your stream, after your Source node, use a Filler node and the CLEM expression **stripctrlchars** to replace the characters.

Related information

- [Common Source Node Tabs](#)
-

Analytic Server Source node

The Analytic Server source enables you to run a stream on Hadoop Distributed File System (HDFS). The information in an Analytic Server data source can come from a variety of places, including:

- Text files on HDFS
- Databases
- HCatalog

Typically, a stream with an Analytic Server Source will be executed on HDFS; however, if a stream contains a node that is not supported for execution on HDFS, then as much of the stream as possible will be "pushed back" to Analytic Server, and then SPSS® Modeler Server will attempt to process the remainder of the stream. You will need to subsample very large datasets; for example, by placing a Sample node within the stream.

If you want to use your own Analytic Server connection instead of the default connection defined by your administrator, deselect Use default Analytic Server and select your connection.

Data source. Assuming you or your SPSS Modeler Server administrator has established a connection, you select a data source containing the data you wish to use. A data source contains the files and metadata associated with that source. Click Select to display a list of available data sources. See the topic [Selecting a data source](#) for more information.

If you need to create a new data source or edit an existing one, click Launch Data Source Editor....

Note that using multiple Analytic Server connections can be useful in controlling the flow of data. For example, when using the Analytic Server Source and Export nodes, you may want to use different Analytic Server connections in different branches of a stream so that when each branch runs it uses its own Analytic Server and no data will be pulled to the IBM® SPSS Modeler Server. Note that if a branch contains more than one Analytic Server connection, the data will be pulled from the Analytic Servers to the IBM SPSS Modeler Server.

- [Selecting a data source](#)
 - [Amending credentials](#)
 - [Supported nodes](#)
-

Selecting a data source

The Data Sources table displays a list of the available data sources. Select the source you want to use and click OK.

Click Show Owner to display the data source owner.

Filter by enables you to filter the data source listing on Keyword, which checks the filter criteria against the data source name and data source description, or Owner. You can enter a combination of string, numeric, or wild card characters described below as filter criteria. The search string is case sensitive. Click Refresh to update the Data Sources table.

- An underscore can be used to represent any single character in the search string.
 - % A percent sign can be used to represent any sequence of zero or more characters in the search string.
-

Amending credentials

If your credentials for accessing Analytic Server are different from the credentials for accessing SPSS® Modeler Server, you will need to enter the Analytic Server credentials when running a stream on Analytic Server. If you do not know your credentials, contact your server administrator.

Supported nodes

Many SPSS® Modeler nodes are supported for execution on HDFS, but there may be some differences in the execution of certain nodes, and some are not currently supported. This topic details the current level of support.

General

- Some characters that are normally acceptable within a quoted Modeler field name will not be accepted by Analytic Server.
- For a Modeler stream to be run in Analytic Server, it must begin with one or more Analytic Server Source nodes and end in a single modeling node or Analytic Server Export node.
- It is recommended that you set the storage of continuous targets as real rather than integer. Scoring models always write real values to the output data files for continuous targets, while the output data model for the scores follows the storage of the target. Thus, if a continuous target has integer storage, there will be a mismatch in the written values and the data model for the scores, and this mismatch will cause errors when you attempt to read the scored data.
- If a field measurement is Geospatial, the function for @OFFSET is not supported.

Source

- A stream that begins with anything other than an Analytic Server source node will be run locally.

Record operations

All Record operations are supported, with the exception of the Streaming TS and Space-Time-Boxes nodes. Further notes on supported node functionality follow.

Select

- Supports the same set of functions supported by the [Derive node](#).

Sample

- Block-level sampling is not supported.
- Complex Sampling methods are not supported.
- First n sampling with "Discard sample" is not supported.
- First n sampling with $N > 20000$ is not supported.
- 1-in- n sampling is not supported when "Maximum sample size" not set.
- 1-in- n sampling is not supported when $N * "Maximum sample size" > 20000$.
- Random % block level sampling is not supported.
- Random % currently supports supplying a seed.

Aggregate

- Contiguous keys are not supported. If you are reusing an existing stream that is set up to sort the data and then use this setting in the Aggregate node, change the stream to remove the Sort node.
- Order statistics (Median, 1st Quartile, 3rd Quartile) are computed approximately, and supported through the Optimization tab.

Sort

- The Optimization tab is not supported.

In a distributed environment, there are a limited number of operations that preserve the record order established by the Sort node.

- A Sort followed by an Export node produces a sorted data source.
- A Sort followed by a Sample node with First record sampling returns the first N records.

In general, you should place a Sort node as close as possible to the operations that need the sorted records.

Merge

- Merge by Order is not supported.
- The Optimization tab is not supported.
- Merge operations are relatively slow. If you have available space in HDFS, it can be much faster to merge your data sources once and use the merged source in following streams than to merge the data sources in each stream.

R Transform

The R syntax in the node should consist of record-at-a-time operations.

Field operations

All Field operations are supported, with the exception of the Anonymize, Transpose, Time Intervals, and History nodes. Further notes on supported node functionality follow.

Auto Data Prep

- Training the node is not supported. Applying the transformations in a trained Auto Data Prep node to new data is supported.

Derive

- All Derive functions are supported, with the exception of sequence functions.
- Deriving a new field as a Count is essentially a sequence operation, and thus not supported.
- Split fields cannot be derived in the same stream that uses them as splits; you will need to create two streams; one that derives the split field and one that uses the field as splits.

Filler

- Supports the same set of functions supported by the [Derive node](#).

Binning

The following functionality is not supported.

- Optimal binning
- Ranks
- Tiles -> Tiling: Sum of values
- Tiles -> Ties: Keep in current and Assign randomly
- Tiles ->Custom N: Values over 100, and any N value where 100 % N is not equal to zero.

RFM Analysis

- The Keep in current option for handling ties is not supported. RFM recency, frequency, and monetary scores will not always match those computed by Modeler from the same data. The score ranges will be the same but score assignments (bin numbers) may differ by one.

Graphs

All Graph nodes are supported.

Modeling

The following Modeling nodes are supported: Time Series, TCM, Isotonic-AS, Extension Model, Tree-AS, C&R Tree, Quest, CHAID, Linear, Linear-AS, Neural Net, GLE, LSVM, TwoStep-AS, Random Trees, STP, Association Rules, XGBoost-AS, Random Forest, and K-Means-AS. Further notes on those nodes' functionality follow.

Linear

When building models on big data, you will typically want to change the objective to Very large datasets, or specify splits.

- Continued training of existing PSM models is not supported.
- The Standard model building objective is only recommended if split fields are defined so that the number of records in each split is not too large, where the definition of "too large" is dependent upon the power of individual nodes in your Hadoop cluster. By contrast, you also need to be careful to ensure that splits are not defined so finely that there are too few records to build a model.
- The Boosting objective is not supported.
- The Bagging objective is not supported.
- The Very large datasets objective is not recommended when there are few records; it will often either not build a model or will build a degraded model.
- Automatic Data Preparation is not supported. This can cause problems when trying to build a model on data with many missing values; normally these would be imputed as part of automatic data preparation. A workaround would be to use a tree model or a neural network with the Advanced setting to impute missing values selected.
- The accuracy statistic is not computed for split models.

Neural Net

When building models on big data, you will typically want to change the objective to Very large datasets, or specify splits.

- Continued training of existing standard or PSM models is not supported.
- The Standard model building objective is only recommended if split fields are defined so that the number of records in each split is not too large, where the definition of "too large" is dependent upon the power of individual nodes in your Hadoop cluster. By contrast, you also need to be careful to ensure that splits are not defined so finely that there are too few records to build a model.
- The Boosting objective is not supported.
- The Bagging objective is not supported.
- The Very large datasets objective is not recommended when there are few records; it will often either not build a model or will build a degraded model.
- When there are many missing values in the data, use the Advanced setting to impute missing values.
- The accuracy statistic is not computed for split models.

C&R Tree, CHAID, and Quest

When building models on big data, you will typically want to change the objective to Very large datasets, or specify splits.

- Continued training of existing PSM models is not supported.
- The Standard model building objective is only recommended if split fields are defined so that the number of records in each split is not too large, where the definition of "too large" is dependent upon the power of individual nodes in your Hadoop cluster. By contrast, you also need to be careful to ensure that splits are not defined so finely that there are too few records to build a model.
- The Boosting objective is not supported.

- The Bagging objective is not supported.
- The Very large datasets objective is not recommended when there are few records; it will often either not build a model or will build a degraded model.
- Interactive sessions is not supported.
- The accuracy statistic is not computed for split models.
- When a split field is present, tree models built locally in Modeler are slightly different from tree models built by Analytic Server, and thus produce different scores. The algorithms in both cases are valid; the algorithms used by Analytic Server are simply newer. Given the fact that tree algorithms tend to have many heuristic rules, the difference between the two components is normal.

Model scoring

All models supported for modeling are also supported for scoring. In addition, locally-built model nuggets for the following nodes are supported for scoring: C&RT, Quest, CHAID, Linear, and Neural Net (regardless of whether the model is standard, boosted bagged, or for very large datasets), Regression, C5.0, Logistic, Genlin, GLMM, Cox, SVM, Bayes Net, TwoStep, KNN, Decision List, Discriminant, Self Learning, Anomaly Detection, Apriori, Carma, K-Means, Kohonen, R, and Text Mining.

- No raw or adjusted propensities will be scored. As a workaround you can get the same effect by manually computing the raw propensity using a Derive node with the following expression: if 'predicted-value' == 'value-of-interest' then 'prob-of-that-value' else 1-'prob-of-that-value' endif

R

The R syntax in the nugget should consist of record-at-a-time operations.

Output

The Matrix, Analysis, Data Audit, Transform, Set Globals, Statistics, Means, and Table nodes are supported. Further notes on supported node functionality follow.

Data Audit

The Data Audit node cannot produce the mode for continuous fields.

Means

The Means node cannot produce a standard error or 95% confidence interval.

Table

The Table node is supported by writing a temporary Analytic Server data source containing the results of upstream operations. The Table node then pages through the contents of that data source.

Export

A stream can begin with an Analytic Server source node and end with an export node other than the Analytic Server export node, but data will move from HDFS to SPSS Modeler Server, and finally to the export location.

Database source node

The Database source node can be used to import data from a variety of other packages using ODBC (Open Database Connectivity), including Microsoft SQL Server, Db2, Oracle, and others.

To read or write to a database, you must have an ODBC data source installed and configured for the relevant database, with read or write permissions as needed. The IBM SPSS Data Access Pack includes a set of ODBC drivers that can be used for this purpose, and these drivers are available from the download site. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

Supported ODBC drivers

For the latest information on which databases and ODBC drivers are supported and tested for use with IBM® SPSS® Modeler, see the product compatibility matrices on the corporate Support site (<http://www.ibm.com/support>).

Where to install drivers

Note: ODBC drivers must be installed and configured on each computer where processing may occur.

- If you are running IBM SPSS Modeler in local (standalone) mode, the drivers must be installed on the local computer.
- If you are running IBM SPSS Modeler in distributed mode against a remote IBM SPSS Modeler Server, the ODBC drivers need to be installed on the computer where IBM SPSS Modeler Server is installed. For IBM SPSS Modeler Server on UNIX systems, see also "Configuring ODBC drivers on UNIX systems" later in this section.
- If you need to access the same data sources from both IBM SPSS Modeler and IBM SPSS Modeler Server, the ODBC drivers must be installed on both computers.
- If you are running IBM SPSS Modeler over Terminal Services, the ODBC drivers need to be installed on the Terminal Services server on which you have IBM SPSS Modeler installed.

Access data from a database

To access data from a database, complete the following steps.

- Install an ODBC driver and configure a data source to the database you want to use.
- In the Database node dialog box, connect to a database using Table mode or SQL Query mode.
- Select a table from the database.
- Using the tabs in the Database node dialog box, you can alter usage types and filter data fields.

More details on the preceding steps are given in the related documentation topics.

Note: If you call database Stored Procedures (SPs) from SPSS Modeler you might see a single output field returned named `RowsAffected` rather than the expected output of the SP. This occurs when ODBC does not return sufficient information to be able to determine the output datamodel of the SP. SPSS Modeler has only limited support for SPs that return output and it is suggested that instead of using SPs you should extract the SELECT from the SP and use either of the following actions.

- Create a view based on the SELECT and choose the view in the Database source node
 - Use the SELECT directly in the Database source node.
- [Setting Database Node Options](#)
 - [Adding a database connection](#)
 - [Potential database issues](#)
 - [Specifying preset values for a database connection](#)
 - [Selecting a Database Table](#)
 - [Querying the database](#)
 - [Using a custom database configuration file](#)

Setting Database Node Options

You can use the options on the Data tab of the Database source node dialog box to gain access to a database and read data from the selected table.

Mode. Select Table to connect to a table using the dialog box controls.

Select SQL Query to query the database selected below using SQL. See the topic [Querying the database](#) for more information.

Data source. For both Table and SQL Query modes, you can enter a name in the Data Source field or select Add new database connection from the drop-down list.

The following options are used to connect to a database and select a table using the dialog box:

Table name. If you know the name of the table you would like to access, enter it in the Table Name field. Otherwise, click the Select button to open a dialog box listing the available tables.

Quote table and column names. Specify whether you want table and column names to be enclosed in quotation marks when queries are sent to the database (if, for example, they contain spaces or punctuation).

- The As needed option will quote table and field names *only* if they include nonstandard characters. Nonstandard characters include non-ASCII characters, space characters, and any non-alphanumeric character other than a full stop (.)
- Select Always if you want *all* table and field names quoted.
- Select Never if you *never* want table and field names quoted.

Strip lead and trail spaces. Select options for discarding leading and trailing spaces in strings.

Note: Comparisons between strings that do and do not use SQL pushback may generate different results where trailing spaces exist.

Reading empty strings from Oracle. When reading from or writing to an Oracle database, be aware that, unlike IBM® SPSS® Modeler and unlike most other databases, Oracle treats and stores empty string values as equivalent to null values. This means that the same data extracted from an Oracle database may behave differently than when extracted from a file or another database, and the data may return different results.

Related information

- [Database source node](#)
- [Adding a database connection](#)
- [Selecting a Database Table](#)
- [Querying the database](#)
- [Using Stream Parameters in an SQL Query](#)

Adding a database connection

To open a database, first select the data source to which you want to connect. On the Data tab, select Add new database connection from the Data Source drop-down list.

This opens the Database Connections dialog box.

Note: For an alternative way to open this dialog, from the main menu, choose: Tools > Databases...

Data sources. Lists the available data sources. Scroll down if you do not see the desired database. Once you have selected a data source and entered any passwords, click Connect. Click Refresh to update the list.

Mode. Select one of the following modes:

- Username and password. If the data source is password protected, enter your user name and the associated password.
- Stored credential. If a credential has been configured in IBM® SPSS® Collaboration and Deployment Services, you can select this option to browse for it in the repository. The credential's user name and password must match the user name and password required to access the database.

Connections. Shows currently connected databases.

- Default. You can optionally choose one connection as the default. Doing so causes Database source or export nodes to have this connection predefined as their data source, though this can be edited if desired.
- Save. Optionally select one or more connections that you want to redisplay in subsequent sessions.
- Data source. The connection strings for the currently connected databases.
- Preset. Indicates (with a * character) whether preset values have been specified for the database connection. To specify preset values, click this column in the row corresponding to the database connection, and choose Specify from the list. See the topic [Specifying preset values for a database connection](#) for more information.

To remove connections, select one from the list and click Remove.

Driver. If you select Driver for the mode instead of Data Source, choose the desired driver from the list.

- In the Attributes field, enter the database connection string. See your database documentation for the proper string to use.
- Enter any display name.
- Enter the database user name and password and click Connect.

To read or write to a database, you must have an ODBC data source installed and configured for the relevant database, with read or write permissions as needed. The IBM SPSS Data Access Pack includes a set of ODBC drivers that can be used for this purpose, and these drivers are available from the download site. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

Supported ODBC drivers

For the latest information on which databases and ODBC drivers are supported and tested for use with IBM SPSS Modeler, see the product compatibility matrices on the corporate Support site (<http://www.ibm.com/support>).

Where to install drivers

Note: ODBC drivers must be installed and configured on each computer where processing may occur.

- If you are running IBM SPSS Modeler in local (standalone) mode, the drivers must be installed on the local computer.
- If you are running IBM SPSS Modeler in distributed mode against a remote IBM SPSS Modeler Server, the ODBC drivers need to be installed on the computer where IBM SPSS Modeler Server is installed. For IBM SPSS Modeler Server on UNIX systems, see also "Configuring ODBC drivers on UNIX systems" later in this section.
- If you need to access the same data sources from both IBM SPSS Modeler and IBM SPSS Modeler Server, the ODBC drivers must be installed on both computers.
- If you are running IBM SPSS Modeler over Terminal Services, the ODBC drivers need to be installed on the Terminal Services server on which you have IBM SPSS Modeler installed.

Configuring ODBC drivers on UNIX systems

By default, the DataDirect Driver Manager is not configured for IBM SPSS Modeler Server on UNIX systems. To configure UNIX to load the DataDirect Driver Manager, enter the following commands:

```
cd <modeler_server_install_directory>/bin  
rm -f libspssodbc.so
```

Then run this command if you want to use the UTF8 driver wrapper:

```
ln -s libspssodbc_datadirect.so libspssodbc.so
```

Or run this command instead if you want to use the UTF16 driver wrapper:

```
ln -s libspssodbc_datadirect_utf16.so libspssodbc.so
```

Doing so removes the default link and creates a link to the DataDirect Driver Manager.

Note: The UTF16 driver wrapper is required to use SAP HANA or IBM Db2 CLI drivers for some databases. DashDB requires the IBM Db2 CLI driver.

To configure SPSS Modeler Server:

1. Configure the SPSS Modeler Server start up script modelersrv.sh to source the IBM SPSS Data Access Pack odbc.sh environment file by adding the following line to modelersrv.sh:

```
. /<pathtoSDAPinstall>/odbc.sh
```

Where <pathtoSDAPinstall> is the full path to your IBM SPSS Data Access Pack installation.

2. Restart SPSS Modeler Server.

In addition, for SAP HANA and IBM Db2 only, add the following parameter definition to the DSN in your odbc.ini file to avoid buffer overflows during connection:

```
DriverUnicodeType=1
```

Note: The libspssodbc_datadirect_utf16.so wrapper is also compatible with the other SPSS Modeler Server supported ODBC drivers.

Potential database issues

Depending on which database you use; there some potential issues that you should be aware of.

IBM Db2

When attempting to cache a node in a stream which reads data from a Db2 database, you may see the following error message:

```
A default table space could not be found with a pagesize of at least 4096 that authorization
ID TEST is authorized to use
```

To configure Db2 to enable in-database caching to work properly in SPSS® Modeler, the database administrator should create a "user temporary" tablespace and grant access to this tablespace to the relevant Db2 accounts.

We recommend using a pagesize of 32768 in the new tablespace, as this will increase the limit on the number of fields that can be successfully cached.

IBM Db2 for z/OS

- Scoring a subset of algorithms, with confidences enabled, using generated SQL can return an error on execution. The issue is specific to Db2 for z/OS; to fix this, use the SPSS Modeler Server Scoring Adapter for Db2 on z/OS.
- When running streams against Db2 for z/OS, you may experience database errors if the timeout for idle database connections is enabled and set too low. In Db2 for z/OS version 8, the default changed from no timeout to 2 minutes. The solution is to increase the value of the Db2 system parameter **IDLE THREAD TIMEOUT (IDTHTOIN)**, or reset the value to 0.

Oracle

When you run a stream that contains an Aggregate node, the values returned for the 1st and 3rd Quartiles, when pushing back SQL to an Oracle database, may differ from those that are returned in native mode.

Specifying preset values for a database connection

For some databases, you can specify a number of default settings for the database connection. The settings all apply to database export.

The database types that support this feature are as follows.

- SQL Server Enterprise and Developer editions. See the topic [Settings for SQL Server](#) for more information.
- Oracle Enterprise or Personal editions. See the topic [Settings for Oracle](#) for more information.

- IBM Db2 for z/OS and Teradata both connect to a database or schema in a similar way. See the topic [Settings for IBM Db2 for z/OS, IBM Db2 LUW, and Teradata](#) for more information.

If you are connected to a database or schema that does not support this feature, you see the message No presets can be configured for this database connection.

- [Settings for SQL Server](#)
- [Settings for Oracle](#)
- [Settings for IBM Db2 for z/OS, IBM Db2 LUW, and Teradata](#)

Settings for SQL Server

These settings are displayed for SQL Server Enterprise and Developer editions.

Use compression. If selected, creates tables for export with compression.

Compression for. Choose the level of compression.

- Row. Enables row-level compression (for example, the equivalent of `CREATE TABLE MYTABLE (...) WITH (DATA_COMPRESSION = ROW)`; in SQL).
- Page. Enables page-level compression (for example, `CREATE TABLE MYTABLE (...) WITH (DATA_COMPRESSION = PAGE)`; in SQL).

Settings for Oracle

Oracle settings - Basic option

These settings are displayed for Oracle Enterprise or Personal editions using the Basic option.

Use compression. If selected, creates tables for export with compression.

Compression for. Choose the level of compression.

- Default. Enables default compression (for example, `CREATE TABLE MYTABLE (...) COMPRESS`; in SQL). In this case, it has the same effect as the Basic option.
- Basic. Enables basic compression (for example, `CREATE TABLE MYTABLE (...) COMPRESS BASIC`; in SQL).

Oracle settings - Advanced option

These settings are displayed for Oracle Enterprise or Personal editions using the Advanced option.

Use compression. If selected, creates tables for export with compression.

Compression for. Choose the level of compression.

- Default. Enables default compression (for example, `CREATE TABLE MYTABLE (...) COMPRESS`; in SQL). In this case, it has the same effect as the Basic option.
- Basic. Enables basic compression (for example, `CREATE TABLE MYTABLE (...) COMPRESS BASIC`; in SQL).
- OLTP. Enables OLTP compression (for example, `CREATE TABLE MYTABLE (...) COMPRESS FOR OLTP`; in SQL).
- Query Low/High. (Exadata servers only) Enables hybrid columnar compression for query (for example, `CREATE TABLE MYTABLE (...) COMPRESS FOR QUERY LOW`; or `CREATE TABLE MYTABLE (...) COMPRESS FOR QUERY HIGH`; in SQL). Compression for query is useful in data warehousing environments; `HIGH` provides a higher compression ratio than `LOW`.
- Archive Low/High. (Exadata servers only) Enables hybrid columnar compression for archive (for example, `CREATE TABLE MYTABLE (...) COMPRESS FOR ARCHIVE LOW`; or `CREATE TABLE MYTABLE (...) COMPRESS FOR ARCHIVE HIGH`; in SQL). Compression for archive is useful for compressing data that will be stored for long periods of time; `HIGH` provides a higher compression ratio than `LOW`.

Settings for IBM Db2 for z/OS, IBM Db2 LUW, and Teradata

When you specify presets for IBM Db2 for z/OS, IBM Db2 LUW, or Teradata, you are prompted to select the following items:

Use server scoring adapter database or Use server scoring adapter schema. If selected, enables the Server scoring adapter database or Server scoring adapter schema option.

Server scoring adapter database or Server scoring adapter schema From the drop-down list, select the connection you require.

In addition, for Teradata, you can also set query banding details to provide additional metadata to assist with items such as workload management; collating, identifying, and resolving queries; and tracking database usage.

Spelling query banding. Choose if the query banding will be set once for the entire time you are working with a Teradata database connection (For Session), or if it will be set every time you run a stream (For Transaction).

Note: If you set query banding on a stream, the banding is lost if you copy that stream to another machine. To prevent this, you can use scripting to run your stream and use the key word *querybanding* in the script to apply the settings you require.

Required database permissions

For SPSS® Modeler database capabilities to function correctly, grant access to the following items to any user IDs used:

Db2 LUW

SYSIBM.SYSDUMMY1
SYSIBM.SYSFOREIGNKEYS
SYSIBM.SYSINDEXES
SYSIBM.SYSKEYCOLUSE
SYSIBM.SYSKEYS
SYSIBM.SYSPARMS
SYSIBM.SYSRELS
SYSIBM.SYSROUTINES
SYSIBM.SYSROUTINES_SRC
SYSIBM.SYSSYNONYMS
SYSIBM.SYSTABCONST
SYSIBM.SYSTABCONSTPKC
SYSIBM.SYSTABLES
SYSIBM.SYSTRIGGERS
SYSIBM.SYSVIEWDEP
SYSIBM.SYSVIEWS
SYSCAT.TABLESPACES
SYSCAT.SCHEMATA

Db2/z

SYSIBM.SYSDUMMY1
SYSIBM.SYSFOREIGNKEYS
SYSIBM.SYSINDEXES
SYSIBM.SYSKEYCOLUSE
SYSIBM.SYSKEYS
SYSIBM.SYSPARMS
SYSIBM.SYSRELS
SYSIBM.SYSROUTINES
SYSIBM.SYSROUTINES_SRC
SYSIBM.SYSSYNONYMS
SYSIBM.SYSTABCONST
SYSIBM.SYSTABLES
SYSIBM.SYSTRIGGERS
SYSIBM.SYSVIEWDEP
SYSIBM.SYSVIEWS
SYSIBM.SYSDUMMYU
SYSIBM.SYSPACKSTMT

Teradata

DBC.Functions
DBC.USERS

Selecting a Database Table

After you have connected to a data source, you can choose to import fields from a specific table or view. From the Data tab of the Database dialog box, you can either enter the name of a table in the Table Name field or click Select to open the Select Table/View dialog box which lists the available tables and views.

Show table owner. Select if a data source requires that the owner of a table must be specified before you can access the table. Deselect this option for data sources that do not have this requirement.

Note: SAS and Oracle databases usually require you to show the table owner.

Tables/Views. Select the table or view to import.

Show. Lists the columns in the data source to which you are currently connected. Click one of the following options to customize your view of the available tables:

- Click User Tables to view ordinary database tables created by database users.
- Click System Tables to view database tables owned by the system (for example, tables that provide information about the database, such as details of indexes). This option can be used to view the tabs used in Excel databases. (Note that a separate Excel source node is also available. See the topic [Excel Source node](#) for more information.)
- Click Views to view virtual tables based on a query involving one or more ordinary tables.
- Click Synonyms to view synonyms created in the database for any existing tables.

Name/Owner filters. These fields allow you to filter the list of displayed tables by name or owner. For example, type `SYS` to list only tables with that owner. For wildcard searches, an underscore (`_`) can be used to represent any single character and a percent sign (`%`) can represent any sequence of zero or more characters.

Set As Default. Saves the current settings as the default for the current user. These settings will be restored in the future when a user opens a new table selector dialog box *for the same data source name and user login only*.

Related information

- [Database source node](#)
- [Setting Database Node Options](#)
- [Adding a database connection](#)
- [Querying the database](#)
- [Using Stream Parameters in an SQL Query](#)

Querying the database

Once you've connected to a data source, you can choose to import fields using SQL queries. From the main dialog box, select SQL Query as the connection mode. This adds a query editor window in the dialog box. Using the query editor, you can create or load one or more SQL queries whose result set will be read into the data stream.

If you specify multiple SQL queries, separate them with semicolons (`;`) and ensure that there's no multiple SELECT statement.

To cancel and close the query editor window, select Table as the connection mode.

You can include SPSS® Modeler stream parameters (a type of user-defined variable) in the SQL query. See [Using Stream Parameters in an SQL Query](#) for more information.

Load Query. Click to open the file browser and load a previously saved query.

Save Query. Click to open the Save Query dialog box to save the current query.

Import Default. Click to import an example SQL

`SELECT` statement constructed automatically using the table and columns selected in the dialog box.

Clear. Clear the contents of the work area. Use this option when you want to start over.

Split text. The default option Never means that the query will be sent to the database as a whole. Alternatively, you can choose As needed, which means that SPSS Modeler attempts to parse the query and identify if there are SQL statements that should be send to the database one after another.

Important: Depending on the database you're using, SPSS Modeler may attempt to run the custom SQL you entered to obtain the data model the SQL results in. For example, when using MySQL or Google BigQuery, by default `SQLExecute()` is called when getting the table schema; so SPSS Modeler will run the SQL and obtain the data model. This isn't the case for most database drivers.

If you want to avoid this, see [Using a custom database configuration file](#) for details about customizing how SPSS Modeler processes SQL in such cases.

- [Using Stream Parameters in an SQL Query](#)

Using Stream Parameters in an SQL Query

When writing an SQL query to import fields, you can include SPSS® Modeler stream parameters that have been previously defined. All types of stream parameter are supported.

The following table shows how some examples of stream parameters will be interpreted in the SQL query.

Table 1. Examples of stream parameters

Stream Parameter Name (example)	Storage	Stream Parameter Value	Interpreted as
PString	String	ss	'ss'
PInt	Integer	5	5
PReal	Real	5.5	5.5
PTime	Time	23:05:01	t{'23:05:01'}
PDate	Date	2011-03-02	d{'2011-03-02'}
PTimestamp	TimeStamp	2011-03-02 23:05:01	ts{'2011-03-02 23:05:01'}
PColumn	Unknown	IntValue	IntValue

In the SQL query, you specify a stream parameter in the same way as in a CLEM expression, namely by '\$P-<parameter_name>', where <parameter_name> is the name that has been defined for the stream parameter.

When referencing a field, the storage type must be defined as Unknown, and the parameter value must be enclosed in quotation marks if needed. Thus, using the examples shown in the table, if you entered the SQL query:

```
select "IntValue" from Table1 where "IntValue" < '$P-PInt';
```

it would be evaluated as:

```
select "IntValue" from Table1 where "IntValue" < 5;
```

If you were to reference the **IntValue** field by means of the **PColumn** parameter, you would need to specify the query as follows to get the same result:

```
select "IntValue" from Table1 where "'$P-PColumn'" < '$P-PInt';
```

Related information

- [Database source node](#)
- [Setting Database Node Options](#)
- [Adding a database connection](#)
- [Selecting a Database Table](#)
- [Querying the database](#)

Using a custom database configuration file

If you need to customize the way SPSS® Modeler processes SQL, you can use a custom database configuration file. This allows for a range of customizations you might need depending on your specific circumstances. Full documentation for this capability isn't available. Contact your database administrator or IBM Support for assistance with your specific needs regarding this capability, but note that IBM doesn't support customizations done via these configuration files that aren't provided by IBM.

To implement a custom database configuration file

1. Create a .cfg file for your specific database.
2. Add options as desired. Extensive customization is possible. Note that the file must be properly named and properly formatted according to your specific database.
3. Place the file in the config folder of your SPSS Modeler Server and/or SPSS Modeler Client installation directory.

The .cfg file name must follow the format odbc-<db>-custom-properties.cfg, where <db> is one of the following database names:

- bigquery
- db2
- greenplum
- hana
- hive
- impala
- informix
- mssql
- mysql
- neoview
- netezza

- oracle
- postgresql
- redshift
- soliddb
- sybase
- teradata
- vertica

Example: execute_while_getting_schema

When using MySQL database with custom SQL in a Database source node, by default the MySQL database driver calls `SQLExecute()` when getting the table schema. As a result, SPSS Modeler needs to run your SQL and obtain the data model. If you don't want SPSS Modeler to obtain the data model, follow these steps:

1. Create a file called odbc-mysql-custom-properties.cfg.
2. Add the following line and set it to `N` to override the default behavior for MySQL:

```
execute_while_getting_schema, N
```

3. Copy the file into the config directory of your SPSS Modeler Server and SPSS Modeler Client installations.

Example: sqlmx_sort_by

By default, `order_by` isn't permitted within nested SQL during SQL pushback. By enabling `order_by`, you can force SQL pushback to work for the Sample node when there's a Sort node ahead of it. For example, to enable `order_by` on Google BigQuery database, follow these steps:

1. Create a file called odbc-bigquery-custom-properties.cfg.
2. Add the following line and set it to `y` to override the default behavior for Google BigQuery:

```
sqlmx_sort_by, Y
```

3. Copy the file into the config directory of your SPSS Modeler Server and SPSS Modeler Client installations.

Example: uda_list_sql_basic and uda_list_sql_parameter

You may want to use custom SQL to retrieve database functions and aggregate functions. For example, on Oracle database, follow these steps:

1. Create a file called odbc-oracle-custom-properties.cfg.
2. Add the following lines to the file:

```
#Define the UDA (database window aggregates) sqls

uda_list_sql_basic, "SELECT '<src_database_name>',OBJECT_NAME,OWNER,'', '' , CASE WHEN OWNER='SYS' THEN
1 ELSE 0 END BUILTIN FROM ALL_ARGUMENTS WHERE OBJECT_ID IN (SELECT OBJECT_ID FROM ALL PROCEDURES WHERE
AGGREGATE = 'YES') AND OBJECT_ID NOT IN (SELECT DISTINCT OBJECT_ID FROM ALL_ARGUMENTS WHERE PLS_TYPE IS
NULL) AND ARGUMENT_NAME IS NULL ORDER BY OBJECT_ID"

uda_list_sql_parameter, "SELECT POSITION, DATA_PRECISION,DATA_SCALE,DATA_TYPE,'',DATA_TYPE,'',0 FROM
ALL_ARGUMENTS WHERE OBJECT_ID IN (SELECT OBJECT_ID FROM ALL PROCEDURES WHERE AGGREGATE = 'YES') AND
OBJECT_ID NOT IN (SELECT DISTINCT OBJECT_ID FROM ALL_ARGUMENTS WHERE PLS_TYPE IS NULL) ORDER BY
OBJECT_ID,POSITION"
```

3. Copy the file into the config directory of your SPSS Modeler Server and SPSS Modeler Client installations.

`uda_list_sql_basic` and `uda_list_sql_parameter` may use other custom SQL provided they conform to the following table schemas below (step 2 is an example).

```
#table schema for uda_list_sql_basic
databaseName,function,schema,catalog,description,isBuiltIn

#table schema for uda_list_sql_parameter
position,precision,scale,returnType,returnTypeName,parameterTypes,parameterTypeNames,isVarArg
```

Variable File Node

You can use Variable File nodes to read data from free-field text files (files whose records contain a constant number of fields but a varied number of characters), also known as delimited text files. This type of node is also useful for files with fixed-length header text and certain types of annotations. Records are read one at a time and passed through the stream until the entire file is read.

Note regarding reading in geospatial data

If the node contains geospatial data, and the node was created as an export from a flat file, you must follow some extra steps to set up the geospatial metadata. For more information, see [Importing geospatial data into the Variable File Node](#).

Notes for reading in delimited text data

- Records must be delimited by a newline character at the end of each line. The newline character must not be used for any other purpose (such as within any field name or value). Leading and trailing spaces should ideally be stripped off to save space, although this is not critical. Optionally these spaces can be stripped out by the node.
- Fields must be delimited by a comma or other character that ideally is used only as a delimiter, meaning it does not appear within field names or values. If this is not possible, then all text fields can be wrapped in double quotation marks, if none of the field names or text values contains a double quotation mark. If field names or values do contain double quotation marks, then text fields can be wrapped in single quotation marks as an alternative, again if single quotation marks are not used elsewhere within values. If neither single or double quotation marks can be used, then text values need to be amended to remove or replace either the delimiter character, or single or double quotation marks.
- Each row, including the header row, should contain the same number of fields.
- The first line should contain the field names. If not, clear Read field names from file to give each field a generic name such as Field1, Field2, and so on.
- The second line must contain the first record of data. There must no blank lines or comments.
- Numeric values must not include the thousands separator or grouping symbol—without the comma in 3,000.00, for example. The decimal indicator (period or full-stop in the US or the UK) must be used only where appropriate.
- Date and time values should be in one of the formats that are recognized in the Stream Options dialog box, such as **DD/MM/YYYY** or **HH:MM:SS**. All dates and time fields in the file should ideally follow the same format, and any field that contains a date must use the same format for all values within that field.
- [Setting options for the Variable File Node](#)
- [Importing geospatial data into the Variable File Node](#)

Setting options for the Variable File Node

You set the options on the File tab of the Variable File node dialog box.

File Specify the name of the file. You can enter a filename or click the ellipsis button (...) to select a file. The file path is shown once you select a file, and its contents are displayed with delimiters in the panel below it.

The sample text that is displayed from your data source can be copied and pasted into the following controls: EOL comment characters and user-specified delimiters. Use Ctrl-C and Ctrl-V to copy and paste.

Read field names from file Selected by default, this option treats the first row in the data file as labels for the column. If your first row is not a header, deselect to automatically give each field a generic name, such as *Field1*, *Field2*, for the number of fields in the dataset.

Specify number of fields Specify the number of fields in each record. The number of fields can be detected automatically as long as the records are new-line terminated. You can also set a number manually.

Skip header characters Specify how many characters you want to ignore at the beginning of the first record.

EOL comment characters Specify characters, such as # or !, to indicate annotations in the data. Wherever one of these characters appears in the data file, everything up to but not including the next new-line character will be ignored.

Strip lead and trail spaces Select options for discarding leading and trailing spaces in strings on import.

Note: Comparisons between strings that do and do not use SQL pushback may generate different results where trailing spaces exist.

Invalid characters Select Discard to remove invalid characters from the data source. Select Replace with to replace invalid characters with the specified symbol (one character only). Invalid characters are null characters or any character that does not exist in the encoding method specified.

Encoding Specifies the text-encoding method used. You can choose between the system default, stream default, or UTF-8.

- The system default is specified in the Windows Control Panel or, if running in distributed mode, on the server computer.
- The stream default is specified in the Stream Properties dialog box.

Decimal symbol Select the type of decimal separator that is used in your data source. The Stream default is the character that is selected from the Options tab of the stream properties dialog box. Otherwise, select either Period (.) or Comma (,) to read all data in this dialog box using the chosen character as the decimal separator.

Line delimiter is newline character To use the newline character as the line delimiter, instead of a field delimiter, select this option. For example, this may be useful if there is an odd number of delimiters in a row that cause the line to wrap. Note that selecting this option means you cannot

select Newline in the Delimiters list.

Note: If you select this option, any blank values at the end of data rows will be stripped out.

Delimiters. Using the check boxes listed for this control, you can specify which characters, such as the comma (,), define field boundaries in the file. You can also specify more than one delimiter, such as ", |" for records that use multiple delimiters. The default delimiter is the comma.

Note: If the comma is also defined as the Decimal symbol, the default settings here will not work. In cases where the comma is both the Field delimiter and the Decimal symbol, select Other in the Field delimiters list. Then manually specify a comma in the entry field.

Select Allow multiple blank delimiters to treat multiple adjacent blank delimiter characters as a single delimiter. For example, if one data value is followed by four spaces and then another data value, this group would be treated as two fields rather than five.

Lines to scan for column and type Specify how many lines and columns to scan for specified data types.

Automatically recognize dates and times To enable IBM® SPSS® Modeler to automatically attempt to recognize data entries as dates or times, select this check box. For example, this means that an entry such as 07-11-1965 will be identified as a date and 02:35:58 will be identified as a time; however, ambiguous entries such as 07111965 or 023558 will show up as integers since there are no delimiters between the numbers.

Note: To avoid potential data problems when you use data files from previous versions of IBM SPSS Modeler, this box is turned off by default for information that is saved in versions prior to 13.

Treat square brackets as lists If you select this check box, the data included between opening and closing square brackets is treated as a single value, even if the content includes delimiter characters such as commas and double quotes. For example, this might include two or three dimensional geospatial data, where the coordinates contained within square brackets are processed as a single list item. For more information, see [Importing geospatial data into the Variable File Node](#)

Quotes. Using the drop-down lists, you can specify how single and double quotation marks are treated on import. You can choose to Discard all quotation marks, Include as text by including them in the field value, or Pair and discard to match pairs of quotation marks and remove them. If a quotation mark is unmatched, you will receive an error message. Both Discard and Pair and discard store the field value (without quotation marks) as a string.

Note: When using Pair and discard, spaces are kept. When using Discard, trailing spaces inside and outside quotes are removed (for example, ' " ab c" , "d ef " , " gh i " ' will result in 'ab c, d ef, gh i'). When using Include as text, quotes are treated as normal characters, so leading and trailing spaces will be stripped naturally.

At any point while you are working in this dialog box, click Refresh to reload fields from the data source. This is useful when you are altering data connections to the source node or when you are working between tabs in the dialog box.

Importing geospatial data into the Variable File Node

If the node contains geospatial data, was created as an export from a flat file, and is used in the same stream in which it was created, the node retains the geospatial metadata and no further configuration steps are needed.

However, if the node is exported and used in a different stream, the geospatial list data is automatically converted to a string format; you must follow some extra steps to restore the list storage type and the associated geospatial metadata.

For more information about lists, see [List storage and associated measurement levels](#).

For more information about the details you can set as geospatial metadata, see [Geospatial measurement sublevels](#).

To set up the geospatial metadata, use the following steps.

1. On the File tab of the Variable File node, select the Treat square brackets as lists check box. Selecting this check box means that the data included between opening and closing square brackets is treated as a single value, even if the content includes delimiter characters such as commas and double quotation marks. Failure to select this check box means that your data is read as a string storage type, any commas in the field are processed as delimiters, and your data structure is interpreted incorrectly.
2. If your data includes single or double quotation marks, select the Pair and discard option in the Single quotes and Double quotes fields, as appropriate.
3. On the Data tab of the Variable File node, for the geospatial data fields, select the Override check box, and change the Storage type from a string to a list.
4. By default, the list Storage type is set as a *List of real* and the underlying value storage type of the list field is set to *Real*. To change the underlying value storage type or the depth, click *Specify...* to display the Storage subdialog box.
5. In the Storage subdialog box you can modify the following settings:
 - Storage Specify the overall storage type of the data field. By default the storage type is set to List; however, the drop-down list contains all other storage types (String, Integer, Real, Date, Time, and Timestamp). If you select any storage type other than List, Value storage and Depth options are unavailable.
 - Value storage Specify the storage types of the elements in the list, as opposed to the field as a whole. When you import geospatial fields, the only relevant storage types are Real and Integer; the default setting is Real.
 - Depth Specify the depth of the list field. The required depth depends on the type of geospatial field and follows these criteria:
 - Point – 0
 - LineString – 1
 - Polygon – 1

- MultiPoint – 1
- MultiLineString – 2
- MultiPolygon – 2

Warning: You must know the type of geospatial field you are converting back into a list and the required depth for that kind of field. If this information is set incorrectly, it is not possible to use the field.

6. On the Types tab of the Variable File node, for the geospatial data field, ensure the Measurement cell contains the correct measurement level. To change the measurement level, in the Measurement cell, click Specify... to display the Values dialog box.
7. In the Values dialog box, the Measurement, Storage, and Depth for the list are displayed. Select the Specify values and labels option and, from the Type drop-down list, select the correct type for the Measurement. Depending on the Type, you might be prompted for more details such as if the data represents 2 or 3 dimensions and which coordinate system is used.

For more information about coordinate systems, see [Setting geospatial options for streams](#).

Fixed File Node

You can use Fixed File nodes to import data from fixed-field text files (files whose fields are not delimited but start at the same position and are of a fixed length). Machine-generated or legacy data are frequently stored in fixed-field format. Using the File tab of the Fixed File node, you can easily specify the position and length of columns in your data.

- [Setting options for the Fixed File node](#)

Setting options for the Fixed File node

The File tab of the Fixed File node enables you to bring data into IBM® SPSS® Modeler and to specify the position of columns and length of records. Using the data preview pane in the center of the dialog box, you can click to add arrows specifying the break points between fields.

File. Specify the name of the file. You can enter a filename or click the ellipsis button (...) to select a file. Once you have selected a file, the file path is shown and its contents are displayed with delimiters in the panel below.

The data preview pane can be used to specify column position and length. The ruler at the top of the preview window helps to measure the length of variables and to specify the break point between them. You can specify break point lines by clicking in the ruler area above the fields. Break points can be moved by dragging and can be discarded by dragging them outside of the data preview region. The ruler is designed to handle ASCII characters.

- Each break-point line automatically adds a new field to the fields table below.
- Start positions indicated by the arrows are automatically added to the Start column in the table below.

Line oriented. Select if you want to skip the new-line character at the end of each record.

Skip header lines. Specify how many lines you want to ignore at the beginning of the first record. This is useful for ignoring column headers.

Record length. Specify the number of characters in each record.

Field. All fields that you have defined for this data file are listed here. There are two ways to define fields:

- Specify fields interactively using the data preview pane above.
- Specify fields manually by adding empty field rows to the table below. Click the button to the right of the fields pane to add new fields. Then, in the empty field, enter a field name, a start position, and a length. These options will automatically add arrows to the data preview pane, which can be easily adjusted.

To remove a previously defined field, select the field in the list and click the red delete button.

Start. Specify the position of the first character in the field. For example, if the second field of a record begins on the sixteenth character, you would enter 16 as the starting point.

Length. Specify how many characters are in the longest value for each field. This determines the cutoff point for the next field.

Strip lead and trail spaces. Select to discard leading and trailing spaces in strings on import.

Note: Comparisons between strings that do and do not use SQL pushback may generate different results where trailing spaces exist.

Invalid characters. Select Discard to remove invalid characters from the data input. Select Replace with to replace invalid characters with the specified symbol (one character only). Invalid characters are null (0) characters or any character that does not exist in the current encoding.

Encoding. Specifies the text-encoding method used. You can choose between the system default, stream default, or UTF-8.

- The system default is specified in the Windows Control Panel or, if running in distributed mode, on the server computer.
- The stream default is specified in the Stream Properties dialog box.

Decimal symbol. Select the type of decimal separator used in your data source. Stream default is the character selected from the Options tab of the stream properties dialog box. Otherwise, select either Period (.) or Comma (,) to read all data in this dialog box using the chosen character as the decimal separator.

Automatically recognize dates and times. To enable IBM SPSS Modeler to automatically attempt to recognize data entries as dates or times, select this check box. For example, this means that an entry such as 07-11-1965 will be identified as a date and 02:35:58 will be identified as a time; however, ambiguous entries such as 07111965 or 023558 will show up as integers since there are no delimiters between the numbers.

Note: To avoid potential data problems when using data files from previous versions of IBM SPSS Modeler, this box is turned off by default for information saved in versions prior to 13.

Lines to scan for type. Specify how many lines to scan for specified data types.

At any point while working in this dialog box, click Refresh to reload fields from the data source. This is useful when altering data connections to the source node or when working between tabs on the dialog box.

Statistics File Node

You can use the Statistics File node to read data directly from a saved IBM® SPSS® Statistics file (.sav or .zsav). This format is now used to replace the cache file from earlier versions of IBM SPSS Modeler. If you would like to import a saved cache file, you should use the IBM SPSS Statistics File node.

Import file. Specify the name of the file. You can enter a filename or click the ellipsis button (...) to select a file. The file path is shown once you have selected a file.

File is password encrypted. Select this box if you know the file is password protected; you are prompted to enter the Password. If the file is password protected, and you do not enter the password, a warning message is displayed if you attempt to change to another tab, refresh the data, preview the node contents, or try to execute a stream containing the node.

Note: Password protected files can only be opened by IBM SPSS Modeler version 16 or greater.

Variable names. Select a method of handling variable names and labels upon import from an IBM SPSS Statistics .sav or .zsav file. Metadata that you choose to include here persists throughout your work in IBM SPSS Modeler and may be exported again for use in IBM SPSS Statistics.

- **Read names and labels.** Select to read both variable names and labels into IBM SPSS Modeler. By default, this option is selected and variable names are displayed in the Type node. Labels may be displayed in charts, model browsers, and other types of output, depending on the options specified in the stream properties dialog box. By default, the display of labels in output is disabled.
- **Read labels as names.** Select to read the descriptive variable labels from the IBM SPSS Statistics .sav or .zsav file rather than the short field names, and use these labels as variable names in IBM SPSS Modeler.

Values. Select a method of handling values and labels upon import from an IBM SPSS Statistics .sav or .zsav file. Metadata that you choose to include here persists throughout your work in IBM SPSS Modeler and may be exported again for use in IBM SPSS Statistics.

- **Read data and labels.** Select to read both actual values and value labels into IBM SPSS Modeler. By default, this option is selected and values themselves are displayed in the Type node. Value labels may be displayed in the Expression Builder, charts, model browsers, and other types of output, depending on the options specified in the stream properties dialog box.
- **Read labels as data.** Select if you want to use the value labels from the .sav or .zsav file rather than the numerical or symbolic codes used to represent the values. For example, selecting this option for data with a gender field whose values of 1 and 2 actually represent *male* and *female*, respectively, will convert the field to a string and import *male* and *female* as the actual values.

It is important to consider missing values in your IBM SPSS Statistics data before selecting this option. For example, if a numeric field uses labels only for missing values (0 = *No Answer*, -99 = *Unknown*), then selecting the option above will import only the value labels *No Answer* and *Unknown* and will convert the field to a string. In such cases, you should import the values themselves and set missing values in a Type node.

Use field format information to determine storage. If this box is cleared, field values that are formatted in the .sav file as integers (i.e., fields specified as Fn.0 in the Variable View in IBM SPSS Statistics) are imported using integer storage. All other field values except strings are imported as real numbers.

If this box is selected (default), all field values except strings are imported as real numbers, whether formatted in the .sav file as integers or not.

Multiple response sets. Any multiple response sets defined in the IBM SPSS Statistics file will automatically be preserved when the file is imported. You can view and edit multiple response sets from any node with a Filter tab. See the topic [Editing Multiple Response Sets](#) for more information.

Data Collection Node

Data Collection source nodes import survey data based on the Survey Reporter Developer Kit that is provided with your Data Collection product. This format distinguishes *case data*—the actual responses to questions gathered during a survey—from the *metadata* that describes how the

case data is collected and organized. Metadata consists of information such as question texts, variable names and descriptions, multiple response variable definitions, translations of the text strings, and the definition of the structure of the case data.

Note: This node requires the Survey Reporter Developer Kit, which is distributed along with your Data Collection product. Aside from installing the Developer Kit, no additional configuration is required.

Comments

- Survey data is read from the flat, tabular VDATA format, or from sources in the hierarchical HDATA format if they include a metadata source.
 - Types are instantiated automatically by using information from the metadata.
 - When survey data is imported into SPSS® Modeler, questions are rendered as fields, with a record for each respondent.
- [Data Collection Import File Options](#)
 - [Data Collection Import Metadata Properties](#)
 - [Database Connection String](#)
 - [Advanced Properties](#)
 - [Importing Multiple Response Sets](#)
 - [Data Collection Column Import Notes](#)

Data Collection Import File Options

The File tab in the Data Collection node enables you to specify options for the metadata and case data you want to import.

Metadata Settings

Note: To see the full list of available provider file types, you need to install the Survey Reporter Developer Kit, available with your Data Collection software product.

Metadata Provider. Survey data can be imported from a number of formats as supported by your Data Collection Survey Reporter Developer Kit. Available provider types include the following:

- DataCollectionMDD. Reads metadata from a questionnaire definition file (.mdd). This is the standard Data Collection Data Model format.
- ADO Database. Reads case data and metadata from ADO files. Specify the name and location of the .adoinfo file that contains the metadata. The internal name of this DSC is *mrADODsc*.
- In2data Database. Reads In2data case data and metadata. The internal name of this DSC is *mrI2dDsc*.
- Data Collection Log File. Reads metadata from a standard Data Collection log file. Typically, log files have a .tmp filename extension. However, some log files may have another filename extension. If necessary, you can rename the file so that it has a .tmp filename extension. The internal name of this DSC is *mrLogDsc*.
- Quancept Definitions File. Converts metadata to Quancept script. Specify the name of the Quancept .qdi file. The internal name of this DSC is *mrQdiDrsDsc*.
- Quanvert Database. Reads Quanvert case data and metadata. Specify the name and location of the .qvinfo or .pkd file. The internal name of this DSC is *mrQvDsc*.
- Data Collection Participation Database. Reads a project's Sample and History Table tables and creates derived categorical variables corresponding to the columns in those tables. The internal name of this DSC is *mrSampleReportingMDSC*.
- Statistics File. Reads case data and metadata from an IBM® SPSS® Statistics .sav file. Writes case data to an IBM SPSS Statistics .sav file for analysis in IBM SPSS Statistics. Writes metadata from an IBM SPSS Statistics .sav file to an .mdd file. The internal name of this DSC is *mrSavDsc*.
- Surveycraft File. Reads SurveyCraft case data and metadata. Specify the name of the SurveyCraft .vq file. The internal name of this DSC is *mrSCDsc*.
- Data Collection Scripting File. Reads from metadata in an mrScriptMetadata file. Typically, these files have an .mdd or .dms filename extension. The internal name of this DSC is *mrScriptMDSC*.
- Triple-S XML File. Reads metadata from a Triple-S file in XML format. The internal name of this DSC is *mrTripleSDsc*.

Metadata properties. Optionally, select Properties to specify the survey version to import as well as the language, context, and label type to use. See the topic [Data Collection Import Metadata Properties](#) for more information.

Case Data Settings

Note: To see the full list of available provider file types, you need to install the Survey Reporter Developer Kit, available with your Data Collection software product.

Get Case Data Settings. When reading metadata from .mdd files only, click Get Case Data Settings to determine what case data sources are associated with the selected metadata, along with the specific settings needed to access a given source. This option is available only for .mdd files.

Case Data Provider. The following provider types are supported:

- ADO Database. Reads case data using the Microsoft ADO interface. Select **OLE-DB UDL** for the case data type, and specify a connection string in the Case Data UDL field. See the topic [Database Connection String](#) for more information. The internal name of this component is *mrADODsc*.
- Delimited Text File (Excel). Reads case data from a comma-delimited (.CSV) file, such as can be output by Excel. The internal name is *mrCsvDsc*.
- Data Collection Data File. Reads case data from a native Data Collection Data Format file. The internal name is *mrDataFileDsc*.
- In2data Database. Reads case data and metadata from an In2data database (.i2d) file. The internal name is *mrI2dDsc*.
- Data Collection Log File. Reads case data from a standard Data Collection log file. Typically, log files have a .tmp filename extension. However, some log files may have another filename extension. If necessary, you can rename the file so that it has a .tmp filename extension. The internal name is *mrLogDsc*.
- Quantum Data File. Reads case data from any Quantum-format ASCII file (.dat). The internal name is *mrPunchDsc*.
- Quancept Data File. Reads case data from a Quancept .drs, .drz, or .dru file. The internal name is *mrQdiDrsDsc*.
- Quanvert Database. Reads case data from a Quanvert qvinfo or .pkd file. The internal name is *mrQvDsc*.
- Data Collection Database (MS SQL Server). Reads case data to a relational Microsoft SQL Server database. See the topic [Database Connection String](#) for more information. The internal name is *mrRdbDsc*.
- Statistics File. Reads case data from an IBM SPSS Statistics .sav file. The internal name is *mrSavDsc*.
- Surveycraft File. Reads case data from a SurveyCraft .qdt file. Both the .vq and .qdt files must be in the same directory, with read and write access for both files. This is not how they are created by default when using SurveyCraft, so one of the files needs to be moved to import SurveyCraft data. The internal name is *mrScDsc*.
- Triple-S Data File. Reads case data from a Triple-S data file, in either fixed-length or comma-delimited format. The internal name is *mrTripleDsc*.
- Data Collection XML. Reads case data from a Data Collection XML data file. Typically, this format may be used to transfer case data from one location to another. The internal name is *mrXmlDsc*.

Case Data Type. Specifies whether case data is read from a file, folder, OLE-DB UDL, or ODBC DSN, and updates the dialog box options accordingly. Valid options depend on the type of provider. For database providers, you can specify options for the OLE-DB or ODBC connection. See the topic [Database Connection String](#) for more information.

Case Data Project. When reading case data from an Data Collection database, you can enter the name of the project. For all other case data types, this setting should be left blank.

Variable Import

Import System Variables. Specifies whether system variables are imported, including variables that indicate interview status (in progress, completed, finish date, and so on). You can choose None, All, or Common.

Import "Codes" Variables. Controls import of variables that represent codes used for open-ended "Other" responses for categorical variables.

Import "SourceFile" Variables. Controls import of variables that contain filenames of images of scanned responses.

Import multi-response variables as. Multiple response variables can be imported as multiple flag fields (a multiple dichotomy set), which is the default method for new streams. Streams created in releases of IBM SPSS Modeler prior to 12.0 imported multiple responses into a single field, with values separate by commas. The older method is still supported to allow existing streams to run as they did previously, but updating older streams to use the new method is recommended. See the topic [Importing Multiple Response Sets](#) for more information.

Related information

- [Data Collection Node](#)
- [Data Collection Import Metadata Properties](#)
- [Database Connection String](#)
- [Advanced Properties](#)
- [Importing Multiple Response Sets](#)
- [Data Collection Column Import Notes](#)

Data Collection Import Metadata Properties

When importing Data Collection survey data, in the Metadata Properties dialog box, you can specify the survey version to import as well as the language, context, and label type to use. Note that only one language, context, and label type can be imported at a time.

Version. Each survey version can be regarded as a snapshot of the metadata used to collect a particular set of case data. As a questionnaire undergoes changes, multiple versions may be created. You can import the latest version, all versions, or a specific version.

- **All versions.** Select this option if you want to use a combination (superset) of all of the available versions. (This is sometimes called a superversion). When there is a conflict between the versions, the most recent versions generally take precedence over the older versions. For example, if a category label differs in any of the versions, the text in the latest version will be used.
- **Latest version.** Select this option if you want to use the most recent version.

- **Specify version.** Select this option if you want to use a particular survey version.

Choosing all versions is useful when, for example, you want to export case data for more than one version and there have been changes to the variable and category definitions that mean that case data collected with one version is not valid in another version. Selecting all of the versions for which you want to export the case data means that generally you can export the case data collected with the different versions at the same time without encountering validity errors due to the differences between the versions. However, depending on the version changes, some validity errors may still be encountered.

Language. Questions and associated text can be stored in multiple languages in the metadata. You can use the default language for the survey or specify a particular language. If an item is unavailable in the specified language, the default is used.

Context. Select the user context you want to use. The user context controls which texts are displayed. For example, select Question to display question texts or Analysis to display shorter texts suitable for displaying when analyzing the data.

Label type. Lists the types of labels that have been defined. The default is label, which is used for question texts in the Question user context and variable descriptions in the Analysis user context. Other label types can be defined for instructions, descriptions, and so forth.

Related information

- [Data Collection Node](#)
 - [Data Collection Import File Options](#)
 - [Database Connection String](#)
 - [Advanced Properties](#)
 - [Importing Multiple Response Sets](#)
 - [Data Collection Column Import Notes](#)
-

Database Connection String

When using the Data Collection node to import case data from a database via an OLE-DB or ODBC, select Edit from the File tab to access the Connection String dialog box, which enables you to customize the connection string passed to the provider in order to fine-tune the connection.

Related information

- [Data Collection Node](#)
 - [Data Collection Import File Options](#)
 - [Data Collection Import Metadata Properties](#)
 - [Advanced Properties](#)
 - [Importing Multiple Response Sets](#)
 - [Data Collection Column Import Notes](#)
-

Advanced Properties

When using the Data Collection node to import case data from a database that requires an explicit login, select Advanced to provide a user ID and password to access the data source.

Related information

- [Data Collection Node](#)
 - [Data Collection Import File Options](#)
 - [Data Collection Import Metadata Properties](#)
 - [Database Connection String](#)
 - [Importing Multiple Response Sets](#)
 - [Data Collection Column Import Notes](#)
-

Importing Multiple Response Sets

Multiple response variables can be imported from Data Collection as multiple dichotomy sets, with a separate flag field for each possible value of the variable. For example, if respondents are asked to select which museums they have visited from a list, the set would include a separate flag field for each museum listed.

After importing the data, you can add or edit multiple response sets from any node that includes a Filter tab. See the topic [Editing Multiple Response Sets](#) for more information.

Importing multiple responses into a single field (for streams created in previous releases)

In older releases of SPSS® Modeler, rather than import multiple responses as described above, they were imported into a single field, with values separate by commas. This method is still supported in order to support for existing streams, but it is recommended that any such streams be updated to use the new method.

Related information

- [Data Collection Node](#)
 - [Data Collection Import File Options](#)
 - [Data Collection Import Metadata Properties](#)
 - [Database Connection String](#)
 - [Advanced Properties](#)
 - [Data Collection Column Import Notes](#)
-

Data Collection Column Import Notes

Columns from the Data Collection data are read into SPSS® Modeler as summarized in the following table.

Table 1. Data Collection column import summary

Data Collection Column Type	SPSS Modeler Storage	Measurement Level
Boolean flag (yes/no)	String	Flag (values 0 and 1)
Categorical	String	Nominal
Date or time stamp	Timestamp	Continuous
Double (floating point value within a specified range)	Real	Continuous
Long (integer value within a specified range)	Integer	Continuous
Text (free text description)	String	Typeless
Level (indicates grids or loops within a question)	Doesn't occur in VDATA and is not imported into SPSS Modeler	
Object (binary data such as a facsimile showing scribbled text or a voice recording)	Not imported into SPSS Modeler	
None (unknown type)	Not imported into SPSS Modeler	
Respondent.Serial column (associates a unique ID with each respondent)	Integer	Typeless

To avoid possible inconsistencies between value labels read from metadata and actual values, all metadata values are converted to lower case. For example, the value label *E1720_years* would be converted to *e1720_years*.

Related information

- [Data Collection Node](#)
 - [Data Collection Import File Options](#)
 - [Data Collection Import Metadata Properties](#)
 - [Database Connection String](#)
 - [Advanced Properties](#)
 - [Importing Multiple Response Sets](#)
-

IBM Cognos source node

The IBM Cognos source node enables you to bring Cognos database data or single list reports into your data mining session. In this way, you can combine the business intelligence features of Cognos with the predictive analytics capabilities of IBM® SPSS® Modeler. You can import relational, dimensionally-modeled relational (DMR), and OLAP data.

From a Cognos server connection, you first select a location from which to import the data or reports. A location contains a Cognos model and all of the folders, queries, reports, views, shortcuts, URLs, and job definitions associated with that model. A Cognos model defines business rules, data descriptions, data relationships, business dimensions and hierarchies, and other administrative tasks.

If you are importing data, you then select the objects that you want to import from the selected package. Objects that you can import include query subjects (which represent database tables) or individual query items (which represent table columns). See [Cognos object icons](#) for more information.

If the package has filters defined, you can import one or more of these. If a filter that you import is associated with imported data, that filter is applied before the data is imported. The data to be imported must be in UTF-8 format.

If you are importing a report, you select a package, or a folder within a package, containing one or more reports. You then select the individual report you want to import. Only single list reports can be imported; multiple lists are not supported.

If parameters have been defined, either for a data object or a report, you can specify values for these parameters before importing the object or report.

Note: The Cognos source node only supports Cognos CQM packages. DQM packages are not supported.

- [Cognos object icons](#)
- [Importing Cognos data](#)
- [Importing Cognos reports](#)
- [Cognos connections](#)
- [Cognos location selection](#)
- [Specifying parameters for data or reports](#)

Cognos object icons

The various types of objects you can import from a Cognos Analytics database are represented by different icons, as the following table illustrates.

Table 1. Cognos object icons

Icon	Object
	Package
	Namespace
	Query subject
	Query item
	Measure dimension
	Measure
	Dimension
	Level hierarchy
	Level
	Filter
	Report
	Standalone calculation

Importing Cognos data

To import data from an IBM Cognos Analytics database, on the Data tab of the IBM Cognos dialog box ensure that Mode is set to Data.

Connection. Click Edit to display a dialog box where you can define the details of a new Cognos connection from which to import data or reports. If you are already logged in to a Cognos server through IBM® SPSS® Modeler, you can also edit the details of the current connection. See [Cognos connections](#) for more information.

Location. When you have established the Cognos server connection, click Edit next to this field to display a list of available packages from which to import content. See [Cognos location selection](#) for more information.

Content. Displays the name of the selected package, together with the namespaces associated with the package. Double-click a namespace to display the objects that you can import. The various object types are denoted by different icons. See [Cognos object icons](#) for more information.

To choose an object to import, select the object and click the upper of the two right arrows to move the object into the Fields to import pane. Selecting a query subject imports all of its query items. Double-clicking a query subject expands it so that you can choose one or more of its

individual query items. You can perform multiple selections with Ctrl-click (select individual items), Shift-click (select a block of items) and Ctrl-A (select all items).

To choose a filter to apply (if the package has filters defined), navigate to the filter in the Content pane, select the filter and click the lower of the two right arrows to move the filter into the Filters to apply pane. You can perform multiple selections with Ctrl-click (select individual filters) and Shift-click (select a block of filters).

Fields to import. Lists the database objects that you have chosen to import into IBM SPSS Modeler for processing. If you no longer require a particular object, select it and click the left arrow to return it to the Content pane. You can perform multiple selections in the same way as for Content.

Filters to apply. Lists the filters that you have chosen to apply to the data before it is imported. If you no longer require a particular filter, select it and click the left arrow to return it to the Content pane. You can perform multiple selections in the same way as for Content.

Parameters. If this button is enabled, the selected object has parameters defined. You can use parameters to make adjustments (for example, perform a parameterized calculation) before importing the data. If parameters are defined but no default is provided, the button displays a warning triangle. Click the button to display the parameters and optionally edit them. If the button is disabled, the report has no parameters defined.

Aggregate data before importing. Check this box if you want to import aggregated data rather than raw data.

Importing Cognos reports

To import a predefined report from an IBM Cognos database, on the Data tab of the IBM Cognos dialog box ensure that Mode is set to Report. Only single list reports can be imported; multiple lists are not supported.

Connection. Click Edit to display a dialog box where you can define the details of a new Cognos connection from which to import data or reports. If you are already logged in to a Cognos server through IBM® SPSS® Modeler, you can also edit the details of the current connection. See [Cognos connections](#) for more information.

Location. When you have established the Cognos server connection, click Edit next to this field to display a list of available packages from which to import content. See [Cognos location selection](#) for more information.

Content. Displays the name of the selected package or folder that contains reports. Navigate to a specific report, select it, and click the right arrow to bring the report into the Report to import field.

Report to import. Indicates the report that you have chosen to import into IBM SPSS Modeler. If you no longer require the report, select it and click the left arrow to return it to the Content pane, or bring a different report into this field.

Parameters. If this button is enabled, the selected report has parameters defined. You can use parameters to make adjustments before importing the report (for example, specifying a start and end date for report data). If parameters are defined but no default is provided, the button displays a warning triangle. Click the button to display the parameters and optionally edit them. If the button is disabled, the report has no parameters defined.

Cognos connections

In the Cognos Connections dialog box, you can select the Cognos Analytics server from which to import or export database objects.

Cognos server URL Type the URL of the Cognos Analytics server from which to import or export. This is the value of the "External dispatcher URI" environment property of IBM Cognos Configuration on the Cognos server. Contact your Cognos system administrator if you are not sure which URL to use.

Mode Select Set Credentials if you want to log in with a specific Cognos namespace, username, and password (for example, as an administrator). Select Use Anonymous connection to log in with no user credentials, in which case you do not complete the other fields.

Alternatively, if you have an IBM Cognos credential that is stored in the IBM® SPSS® Collaboration and Deployment Services Repository, you can use this credential instead of entering user name and password information, or creating an anonymous connection. To use an existing credential, select Stored Credentials and either enter the Credential Name or browse for it.

The Cognos Namespace is modeled by a domain in IBM SPSS Collaboration and Deployment Services.

Namespace ID Specify the Cognos security authentication provider that is used to log on to the server. The authentication provider is used to define and maintain users, groups, and roles, and to control the authentication process. Note this is the Namespace ID, not the Namespace Name (the ID is not always the same as the Name).

User name Enter the Cognos user name with which to log on to the server.

Password Enter the password that is associated with the specified user name.

Save as Default Click this button to store these settings as your default, to avoid having to reenter them every time you open the node.

Cognos location selection

The Specify Location dialog box enables you to select a Cognos package from which to import data, or a package or folder from which to import reports.

Public Folders. If you are importing data, this lists the packages and folders available from the chosen server. Select the package you want to use and click OK. You can select only one package per Cognos source node.

If you are importing reports, this lists the folders and packages containing reports that are available from the chosen server. Select a package or report folder and click OK. You can select only one package or report folder per Cognos source node, though report folders can contain other report folders as well as individual reports.

Specifying parameters for data or reports

If parameters have been defined in Cognos Analytics, either for a data object or a report, you can specify values for these parameters before importing the object or report. An example of parameters for a report would be start and end dates for the report contents.

Name. The parameter name as it is specified in the Cognos database.

Type. A description of the parameter.

Value. The value to assign to the parameter. To enter or edit a value, double-click its cell in the table. Values are not validated here, so any invalid values are detected at run time.

Automatically remove invalid parameters from table. This option is selected by default and will remove any invalid parameters found within the data object or report.

IBM Cognos TM1 Source Node

The IBM Cognos TM1 source node enables you to bring Cognos TM1 data into your data mining session. In this way, you can combine the enterprise planning features of Cognos with the predictive analytics capabilities of IBM® SPSS® Modeler. You can import a flattened version of the multidimensional OLAP cube data.

Note: The TM1 user needs the following permissions: Write privilege of cubes, Read privilege of dimensions, and Write privilege of dimension elements. In addition, IBM Cognos TM1 10.2 Fix Pack 3, or later, is required before SPSS Modeler can import and export Cognos TM1 data. Existing streams that were based on previous versions will still function.

Administrator credentials are not required for this node. But if you are still using the old legacy pre-17.1 TM1 node, administrator credentials are still required.

SPSS Modeler only supports working with Cognos TM1 via IntegratedSecurityMode 1, 4, and 5.

You need to modify the data in TM1 before the data is imported. The data to be imported must be in UTF-8 format.

From an IBM Cognos TM1 administration host connection, you first select a TM1 server from which to import the data; a server contains one or more TM1 cubes. You then select the required cube and within the cube you select the columns and rows you want to import.

Note: Before you can use the TM1 Source or Export nodes in SPSS Modeler, you must verify some settings in the tm1s.cfg file; this is the TM1 server configuration file in the root directory of the TM1 server.

- HTTPPortNumber - set a valid port number; typically 1-65535. Note that this is not the port number you subsequently specify in the connection in the node; it is an internal port used by TM1 that is disabled by default. If necessary, contact your TM1 administrator to confirm the valid setting for this port.
- UseSSL - if you set this to *True*, HTTPS is used as the transport protocol. In this case you must import the TM1 certification to the SPSS Modeler Server JRE.
- [Importing IBM Cognos TM1 data](#)

Importing IBM Cognos TM1 data

To import data from an IBM Cognos TM1 database, on the Data tab of the IBM Cognos TM1 dialog box, specify the server connection details and select the cube and data details.

Note: Before importing data you must carry out some preprocessing within TM1 to ensure the data is in a format that's recognizable to IBM® SPSS® Modeler. This involves filtering your data using the Subset Editor to get the view into the correct size and shape for import.

Note that zero (0) values imported from TM1 will be treated as "null" values (TM1 does not distinguish between blank and zero values). Also, note that non-numeric data (or metadata) from *regular dimensions* can be imported into IBM SPSS Modeler. But importing non-numeric *measures* is not currently supported.

Connection Type. Select Admin Server or TM1 Server. Note that the Admin Server has been removed from [Planning Analytics on Cloud](#), so if you have old streams that connect to an old Admin Server, you can modify them to point to Planning Analytics on Cloud instead. If you select Admin Server here, then you must enter the URL for the server (the host name of the REST API) and the name of the server. If you select TM1 Server, proceed to the following sections.

TM1 Server URL. Type the URL of the administration host where the TM1 server you want to connect to is installed. The administration host is defined as a single URL for all TM1 servers. From this URL, all IBM Cognos TM1 servers installed and running on your environment can be discovered and accessed. Click Login. If you have not previously connected to this server you are prompted to enter your User name and Password; alternatively, you can search for previously entered login details that you have saved as a Stored Credential.

Select a TM1 cube view to import. Displays the name of the cubes within the TM1 server from which you can import data. Double-click a cube to display the view data you can import.

Note:

Only cubes with a dimension can be imported into IBM SPSS Modeler.

If an alias has been defined for an element in your TM1 cube (for example, if a value of 23277 has an alias of **Sales**), the value will be imported – not the alias.

To choose data to import, select the view and click the right arrow to move it into the View to import field. If the view you require isn't visible, double-click a cube to expand its list of views. You can select a public or a private view.

Row dimension(s). Lists the name of the row dimension in the data that you have chosen to import. Scroll through the list of levels and select the one you require.

Column dimension. Lists the name of the column dimension in the data that you have chosen to import. Scroll through the list of levels and select the one you require.

Context dimension(s). Display only. Shows the context dimensions that relate to the selected columns and rows.

TWC source node

The TWC source node imports weather data from The Weather Company, an IBM Business. You can use it to obtain historical or forecast weather data for a location. This can help you develop weather-driven business solutions for better decision-making using the most accurate and precise weather data available.

With this node, you can input weather-related data such as `latitude`, `longitude`, `time`, `day_ind` (indicates night or day), `temp`, `dewpt` (dew point), `rh` (relative humidity), `feels_like` temperature, `heat_index`, `wc` (wind chill), `wx_phrase` (mostly cloudy, partly cloudy, etc), `pressure`, `clds` (clouds), `vis` (visibility), `wspd` (wind speed), `gust`, `wdir` (wind direction), `wdir_cardinal` (NW, NNW, N, etc), `uv_index` (ultraviolet index), and `uv_desc` (low, high, etc).

The TWC source node uses the following APIs:

- TWC Historical Observations Airport (<http://goo.gl/DplOKj>) for historical weather data
- TWC Hourly Forecast (<http://goo.gl/IJjhvZ>) for forecast weather data

Location

Latitude. Enter the latitude value of the location you wish to obtain weather data for, in the format [-90.0~90.0].

Longitude. Enter the longitude value of the location you wish to obtain weather data for, in the format [-180.0~180.0].

Misc

License Key. A license key is required. Enter the license key you obtained from The Weather Company. If you don't have a key, contact your administrator or your IBM representative.

Instead of issuing the key to all users, your administrator may have instead specified the key in a new config.cfg file on the IBM® SPSS® Modeler Server, in which case you can leave this field blank. If specified in both locations, the key in this dialog takes precedence. Note to administrators:

To add the license key on the server, create a new file called config.cfg with the contents `LicenseKey=<LICENSEKEY>` (where `<LICENSEKEY>` is the license key) at the location `<ModelerServerInstallation>\ext\bin\pasw.twcdata`.

Units. Select the measurement unit to use: English, Metric, or Hybrid. The default is Metric.

Time Format

UTC. Select UTC time format if you're importing historical weather data and you don't want SPSS Modeler to access the TWC Hourly Forecast API. Your license key only needs privileges to access the TWC Historical Observations Airport API if you select this option.

Local. Select local time format if you need SPSS Modeler to access the TWC Hourly Forecast API to convert the time from UTC time to local time. Your license key needs privileges to access both TWC APIs if you select this option.

Data Type

Historical. If you want to import historical weather data, select Historical and then specify a start and end date in the format `YYYYMMDD` (for example, `20120101` for January 1, 2012).

Forecast. If you want to import forecast weather data, select Forecast and then specify the hours to forecast.

SAS Source Node

This feature is available in SPSS® Modeler Professional and SPSS Modeler Premium.

The SAS source node enables you to bring SAS data into your data mining session. You can import four types of files:

- SAS for Windows/OS2 (.sd2)
- SAS for UNIX (.ssd)
- SAS Transport File (.tpt)
- SAS version 7/8/9 (.sas7bdat)

When the data is imported, all variables are kept and no variable types are changed. All cases are selected.

- [Setting Options for the SAS Source Node](#)
-

Setting Options for the SAS Source Node

This feature is available in SPSS® Modeler Professional and SPSS Modeler Premium.

Import. Select which type of SAS file to transport. You can choose SAS for Windows/OS2 (.sd2), SAS for UNIX (.SSD), SAS Transport File (.tpt), or SAS Version 7/8/9 (.sas7bdat).

Import file. Specify the name of the file. You can enter a filename or click the ellipsis button (...) to browse to the file's location.

Member. Select a member to import from the SAS transport file selected above. You can enter a member name or click Select to browse through all members in the file.

Read user formats from a SAS data file. Select to read user formats. SAS files store data and data formats (such as variable labels) in different files. Most often, you will want to import the formats as well. If you have a large dataset, however, you may want to deselect this option to save memory.

Format file. If a format file is required, this text box is activated. You can enter a filename or click the ellipsis button (...) to browse to the file's location.

Variable names. Select a method of handling variable names and labels upon import from a SAS file. Metadata that you choose to include here persists throughout your work in IBM® SPSS Modeler and may be exported again for use in SAS.

- **Read names and labels.** Select to read both variable names and labels into IBM SPSS Modeler. By default, this option is selected and variable names are displayed in the Type node. Labels may be displayed in the Expression Builder, charts, model browsers, and other types of output, depending on the options specified in the stream properties dialog box.
- **Read labels as names.** Select to read the descriptive variable labels from the SAS file rather than the short field names and use these labels as variable names in IBM SPSS Modeler.
- [Selecting a Member](#)

Related information

- [SAS Source Node](#)
-

Selecting a Member

This feature is available in SPSS® Modeler Professional and SPSS Modeler Premium.

Once you have specified a SAS transport file, you need to select which member to read into IBM® SPSS Modeler. This dialog box enables you to browse through all members available from the transport file.

Excel Source node

The Excel source node enables you to import data from Microsoft Excel in the .xlsx file format.

File type. Select the Excel file type that you are importing.

Import file. Specifies the name and location of the spreadsheet file to import.

Use Named Range. Enables you to specify a named range of cells as defined in the Excel worksheet. Click the ellipses button (...) to choose from the list of available ranges. If a named range is used, other worksheet and data range settings are no longer applicable and are disabled as a result.

Choose worksheet. Specifies the worksheet to import, either by index or by name.

- By index. Specify the index value for the worksheet you want to import, beginning with 0 for the first worksheet, 1 for the second worksheet, and so on.
- By name. Specify the name of the worksheet you want to import. Click the ellipses button (...) to choose from the list of available worksheets.

Range on worksheet. You can import data beginning with the first non-blank row or with an explicit range of cells.

- Range starts on first non-blank row. Locates the first non-blank cell and uses this as the upper left corner of the data range.
- Explicit range of cells. Enables you to specify an explicit range by row and column. For example, to specify the Excel range A1:D5, you can enter A1 in the first field and D5 in the second (or alternatively, R1C1 and R5C4). All rows in the specified range are returned, including blank rows.

On blank rows. If more than one blank row is encountered, you can choose whether to Stop reading, or choose Return blank rows to continue reading all data to the end of the worksheet, including blank rows.

First row has column names. Indicates that the first row in the specified range should be used as field (column) names. If not selected, field names are generated automatically.

Lines to scan for column and type. You can increase this value if you want IBM® SPSS® Modeler to scan more rows of your Excel data to determine the column type and storage type. The default is 200 rows. Note that this setting can impact performance.

Field Storage and Measurement Level

When reading values from Excel, fields with numeric storage are read in with a measurement level of *Continuous* by default, and string fields are read in as *Nominal*. You can manually change the measurement level (continuous versus nominal) on the Type tab, but the storage is determined automatically (although it can be changed using a conversion function, such as `to_integer`, in a Filler node or Derive node if necessary). See the topic [Setting Field Storage and Formatting](#) for more information.

By default, fields with a mix of numeric and string values read in as numbers, which means that any string values will be set to null (system missing) values in IBM SPSS Modeler. This happens because, unlike Excel, IBM SPSS Modeler does not allow mixed storage types within a field. To avoid this, you can manually set the cell format to Text in the Excel spreadsheet, which causes all values (including numbers) to read in as strings.

XML Source Node

This feature is available in SPSS® Modeler Professional and SPSS Modeler Premium.

Use the XML source node to import data from a file in XML format into an IBM® SPSS Modeler stream. XML is a standard language for data exchange, and for many organizations it is the format of choice for this purpose. For example, a government tax agency might want to analyze data from tax returns that have been submitted online and which have their data in XML format (see <http://www.w3.org/standards/xml/>).

Importing XML data into an IBM SPSS Modeler stream enables you to perform a wide range of predictive analytics functions on the source. The XML data is parsed into a tabular format in which the columns correspond to the different levels of nesting of the XML elements and attributes. The XML items are displayed in XPath format (see <http://www.w3.org/TR/xpath20/>).

Important: The XML source node does not consider namespace declaration. So, for example, your XML files cannot contain a colon (:) character in the **name** tag. If they do, you'll receive errors during execution about invalid characters.

Read a single file. By default, SPSS Modeler reads a single file, which you specify in the XML data source field.

Read all XML files in a directory. Choose this option if you want to read all the XML files in a particular directory. Specify the location in the Directory field that appears. Select the Include subdirectories check box to additionally read XML files from all the subdirectories of the specified directory.

XML data source. Type the full path and file name of the XML source file you want to import, or use the Browse button to find the file.

XML schema. (Optional) Specify the full path and file name of an XSD or DTD file from which to read the XML structure, or use the Browse button to find this file. If you leave this field blank, the structure is read from the XML source file. An XSD or DTD file can have more than one root element. In this case, when you change the focus to a different field, a dialog is displayed where you choose the root element you want to use. See the topic [Selecting from Multiple Root Elements](#) for more information.

Note: XSD Indicators are ignored by SPSS Modeler

XML structure. A hierarchical tree showing the structure of the XML source file (or the schema, if you specified one in the XML schema field). To define a record boundary, select an element and click the right-arrow button to copy the item to the Records field.

Display attributes. Displays or hides the attributes of the XML elements in the XML structure field.

Records (XPath expression). Shows the XPath syntax for an element copied from the XML structure field. This element is then highlighted in the XML structure, and defines the record boundary. Each time this element is encountered in the source file, a new record is created. If this field is empty, the first child element under the root is used as the record boundary.

Read all data. By default, all data in the source file is read into the stream.

Specify data to read. Choose this option if you want to import individual elements, attributes or both. Choosing this option enables the Fields table where you can specify the data you want to import.

Fields. This table lists the elements and attributes selected for import, if you have selected the Specify data to read option. You can either type the XPath syntax of an element or attribute directly into the XPath column, or select an element or attribute in the XML structure and click the right-arrow button to copy the item into the table. To copy all the child elements and attributes of an element, select the element in the XML structure and click the double-arrow button.

- XPath. The XPath syntax of the items to be imported.
- Location. The location in the XML structure of the items to be imported. Fixed path shows the path of the item relative to the element highlighted in the XML structure (or the first child element under the root, if no element is highlighted). Any location denotes an item of the given name at any location in the XML structure. Custom is displayed if you type a location directly into the XPath column.
- [Selecting from Multiple Root Elements](#)
- [Removing Unwanted Spaces from XML Source Data](#)

Selecting from Multiple Root Elements

This feature is available in SPSS® Modeler Professional and SPSS Modeler Premium.

While a properly formed XML file can only have a single root element, an XSD or DTD file can contain multiple roots. If one of the roots matches that in the XML source file, that root element is used, otherwise you need to select one to use.

Choose the root to display. Select the root element you want to use. The default is the first root element in the XSD or DTD structure.

Removing Unwanted Spaces from XML Source Data

This feature is available in SPSS® Modeler Professional and SPSS Modeler Premium.

Line breaks in the XML source data may be implemented by a [CR] [LF] character combination. In some cases these line breaks can occur in the middle of a text string, for example:

```
<description>An in-depth look at creating applications[CR][LF]
with XML.</description>
```

These line breaks may not be visible when the file is opened in some applications, for example a Web browser. However, when the data are read into the stream through the XML source node, the line breaks are converted to a series of space characters.

You can correct this by using a Filler node to remove these unwanted spaces:

Here is an example of how you can achieve this:

1. Attach a Filler node to the XML source node.
2. Open the Filler node and use the field chooser to select the field with the unwanted spaces.
3. Set Replace to Based on condition and set Condition to true.
4. In the Replace with field, enter `replace(" ", "", @FIELD)` and click OK.
5. Attach a Table node to the Filler node and run the stream.

In the Table node output, the text now appears without the additional spaces.

User Input Node

The User Input node provides an easy way for you to create synthetic data--either from scratch or by altering existing data. This is useful, for example, when you want to create a test dataset for modeling.

Creating Data from Scratch

The User Input node is available from the Sources palette and can be added directly to the stream canvas.

1. Click the Sources tab of the nodes palette.
2. Drag and drop or double-click to add the User Input node to the stream canvas.
3. Double-click to open its dialog box and specify fields and values.

Note: User Input nodes that are selected from the Sources palette will be completely blank, with no fields and no data information. This enables you to create synthetic data entirely from scratch.

Generating Data from an Existing Data Source

You can also generate a User Input node from any nonterminal node in the stream:

1. Decide at which point in the stream you want to replace a node.
2. Right-click on the node that will feed its data into the User Input node and choose Generate User Input Node from the menu.
3. The User Input node appears with all downstream processes attached to it, replacing the existing node at that point in your data stream. When generated, the node inherits all of the data structure and field type information (if available) from the metadata.

Note: If data have not been run through all nodes in the stream, then the nodes are not fully instantiated, meaning that storage and data values may not be available when replacing with a User Input node.

- [Setting Options for the User Input Node](#)
-

Setting Options for the User Input Node

The dialog box for a User Input node contains several tools you can use to enter values and define the data structure for synthetic data. For a generated node, the table on the Data tab contains field names from the original data source. For a node added from the Sources palette, the table is blank. Using the table options, you can perform the following tasks:

- Add new fields using the Add a New Field button at the right in the table.
- Rename existing fields.
- Specify data storage for each field.
- Specify values.
- Change the order of fields on the display.

Entering Data

For each field, you can specify values or insert values from the original dataset using the value picker button to the right of the table. See the rules described below for more information on specifying values. You can also choose to leave the field blank--fields left blank are filled with the system null (`$null$`).

To specify string values, simply type them in the Values column, separated by spaces:

Fred Ethel Martin

Strings that include spaces can be wrapped in double-quotes:

```
"Bill Smith" "Fred Martin" "Jack Jones"
```

For numeric fields, you can either enter multiple values in the same manner (listed with spaces between):

```
10 12 14 16 18 20
```

Or you can specify the same series of numbers by setting its limits (10, 20) and the steps in between (2). Using this method, you would type:

```
10,20,2
```

These two methods can be combined by embedding one within the other, such as:

```
1 5 7 10,20,2 21 23
```

This entry will produce the following values:

```
1 5 7 10 12 14 16 18 20 21 23
```

Date and time values can be entered using the current default format selected in the Stream Properties dialog box, for example:

```
11:04:00 11:05:00 11:06:00
```

```
2007-03-14 2007-03-15 2007-03-16
```

For timestamp values, which have both a date and time component, double-quotes must be used:

```
"2007-03-14 11:04:00" "2007-03-14 11:05:00" "2007-03-14 11:06:00"
```

For additional details see comments on data storage below.

Generate data. Enables you to specify how the records are generated when you run the stream.

- **All combinations.** Generates records containing every possible combination of the field values, so each field value will appear in several records. This can sometimes generate more data than is wanted, so often you might follow this node with a sample node.
- **In order.** Generates records in the order in which the data field values are specified. Each field value only appears in one record. The total number of records is equal to the largest number of values for a single field. Where fields have fewer than the largest number, undefined (\$null\$) values are inserted.

Show example

For example, the following entries will generate the records listed in the two following table examples.

- **Age.** 30, 60, 10
- **BP.** LOW
- **Cholesterol.** NORMAL HIGH
- **Drug.** (left blank)

Table 1. Generate data field set to All combinations

Age	BP	Cholesterol	Drug
30	LOW	NORMAL	\$null\$
30	LOW	HIGH	\$null\$
40	LOW	NORMAL	\$null\$
40	LOW	HIGH	\$null\$
50	LOW	NORMAL	\$null\$
50	LOW	HIGH	\$null\$
60	LOW	NORMAL	\$null\$
60	LOW	HIGH	\$null\$

Table 2. Generate data field set to In order

Age	BP	Cholesterol	Drug
30	LOW	NORMAL	\$null\$
40	\$null\$	HIGH	\$null\$
50	\$null\$	\$null\$	\$null\$
60	\$null\$	\$null\$	\$null\$

Data Storage

Storage describes the way data are stored in a field. For example, a field with values of 1 and 0 stores integer data. This is distinct from the measurement level, which describes the usage of the data, and does not affect storage. For example, you may want to set the measurement level for an integer field with values of 1 and 0 to *Flag*. This usually indicates that 1 = *True* and 0 = *False*. While storage must be determined at the source, measurement level can be changed using a Type node at any point in the stream. See the topic [Measurement levels](#) for more information.

Available storage types are:

- String Used for fields that contain non-numeric data, also called alphanumeric data. A string can include any sequence of characters, such as *fred*, *Class 2*, or *1234*. Note that numbers in strings cannot be used in calculations.
- Integer A field whose values are integers.
- Real Values are numbers that may include decimals (not limited to integers). The display format is specified in the Stream Properties dialog box and can be overridden for individual fields in a Type node (Format tab).
- Date Date values specified in a standard format such as year, month, and day (for example, 2007–09–26). The specific format is specified in the Stream Properties dialog box.
- Time Time measured as a duration. For example, a service call lasting 1 hour, 26 minutes, and 38 seconds might be represented as 01:26:38, depending on the current time format as specified in the Stream Properties dialog box.
- Timestamp Values that include both a date and time component, for example 2007–09–26 09:04:00, again depending on the current date and time formats in the Stream Properties dialog box. Note that timestamp values may need to be wrapped in double-quotes to ensure they are interpreted as a single value rather than separate date and time values. (This applies for example when entering values in a User Input node.)
- List Introduced in SPSS® Modeler version 17, along with new measurement levels of Geospatial and Collection, a List storage field contains multiple values for a single record. There are list versions of all of the other storage types.

Table 3. List storage type icons

Icon	Storage type
[A]	List of string
[I]	List of integer
[#]	List of real
[C]	List of time
[D]	List of date
[T]	List of timestamp
[L]	List with a depth greater than zero

In addition, for use with the Collection measurement level, there are list versions of the following measurement levels.

Table 4. List measurement level icons

Icon	Measurement level
[P]	List of continuous
[C]	List of categorical
[F]	List of flags
[N]	List of nominal
[O]	List of ordinal

Lists can be imported into SPSS Modeler in one of three source nodes (Analytic Server, Geospatial, or Variable File), or created within your streams through use of the Derive or Filler field operation nodes.

For more information on Lists and their interaction with the Collection and Geospatial measurement levels, see [List storage and associated measurement levels](#)

Storage conversions. You can convert storage for a field using a variety of conversion functions, such as `to_string` and `to_integer`, in a Filler node. See the topic [Storage Conversion Using the Filler Node](#) for more information. Note that conversion functions (and any other functions that require a specific type of input such as a date or time value) depend on the current formats specified in the Stream Properties dialog box. For example, if you want to convert a string field with values *Jan 2018*, *Feb 2018*, (and so forth) to date storage, select MON YYYY as the default date format for the stream. Conversion functions are also available from the Derive node, for temporary conversion during a derive calculation. You can also use the Derive node to perform other manipulations, such as recoding string fields with categorical values. See the topic [Recoding Values with the Derive Node](#) for more information.

Reading in mixed data. Note that when reading in fields with numeric storage (either integer, real, time, timestamp, or date), any non-numeric values are set to null or system missing. This is because unlike some applications, IBM® SPSS Modeler does not allow mixed storage types within a field. To avoid this, any fields with mixed data should be read in as strings, either by changing the storage type in the source node or in the external application as necessary.

Note: Generated User Input nodes may already contain storage information garnered from the source node if instantiated. An uninstantiated node does not contain storage or usage type information.

Rules for Specifying Values

For symbolic fields, you should leave spaces between multiple values, such as:

HIGH MEDIUM LOW

For numeric fields, you can either enter multiple values in the same manner (listed with spaces between):

10 12 14 16 18 20

Or you can specify the same series of numbers by setting its limits (10, 20) and the steps in between (2). Using this method, you would type:

10,20,2

These two methods can be combined by embedding one within the other, such as:

1 5 7 10,20,2 21 23

This entry will produce the following values:

1 5 7 10 12 14 16 18 20 21 23

Related information

- [User Input Node](#)

Simulation Generate Node

The Simulation Generate node provides an easy way to generate simulated data, either without historical data using user specified statistical distributions, or automatically using the distributions obtained from running a Simulation Fitting node on existing historical data. Generating simulated data is useful when you want to evaluate the outcome of a predictive model in the presence of uncertainty in the model inputs.

Creating data without historical data

The Simulation Generate node is available from the Sources palette and can be added directly to the stream canvas.

1. Click the Sources tab of the nodes palette.
2. Drag and drop or double-click to add the Simulation Generate node to the stream canvas.
3. Double-click to open its dialog box and specify fields, storage types, statistical distributions and distribution parameters.

Note: Simulation Generate nodes that are selected from the Sources palette will be completely blank, with no fields and no distribution information. This enables you to entirely create simulated data without historical data.

Generating simulated data using existing historical data

A Simulation Generate node can also be created by executing a Simulation Fitting terminal node:

1. Right-click on the Simulation Fitting node and choose Run from the menu.
2. The Simulation Generate node appears on the stream canvas with an update link to the Simulation Fitting node.
3. When generated, the Simulation Generate node inherits all of the fields, storage types, and statistical distribution information from the Simulation Fitting node.

Defining an update link to a simulation fitting node

You can create a link between a Simulation Generate node and a Simulation Fitting node. This is useful if you want to update one or more fields with the information of the best fitting distribution, determined by fitting to historical data.

1. Right click on the Simulation Generate node.
2. From the menu, select Define Update Link. The cursor changes to the Link cursor.
3. Click on another node. If this node is a Simulation Fitting node, a link is made. If this node is not a Simulation Fitting node, no link is made, and the cursor changes back to a normal cursor.

If the fields in the Simulation Fitting node are different to those in the Simulation Generate node, a message is displayed informing you that there is a difference.

When the Simulation Fitting node is used to update the linked Simulation Generate node, the result depends on whether the same fields are present in both nodes, and if the fields are unlocked in the Simulation Generate node. The results of updating a Simulation Fitting node are shown in the following table.

Table 1. Results of updating a Simulation Fitting node

	Field in Simulation	Fitting node
Field in Simulation Generate node	Present	Missing
Present and unlocked.	Field is overwritten.	Field is deleted.
Missing.	Field is added.	No change.
Present and locked.	The distribution of the field is not overwritten. The information in the Fit Details dialog box and the correlations are updated.	The field is not overwritten. The correlations are set to zero.
Do not clear Min/Max when refitting check box is selected.	The field is overwritten, apart from the values in the Min, Max column.	

	Field in Simulation	Fitting node
Do not recalculate correlations when refitting check box is selected.	If the field is unlocked, it is overwritten.	The correlations are not overwritten.

Removing an update link to a simulation fitting node

You can remove a link between a Simulation Generate node and a Simulation Fitting node by taking the following steps:

1. Right click on the Simulation Generate node.
2. From the menu, select Remove Update Link. The link is removed.

- [Setting Options for the Simulation Generate Node](#)
- [Clone Field](#)
- [Fit Details](#)
- [Specify Parameters](#)
- [Distributions](#)

Setting Options for the Simulation Generate Node

You can use the options on the Data tab of the Simulation Generate node dialog box to do the following:

- View, specify, and edit the statistical distribution information for the fields.
- View, specify, and edit the correlations between the fields.
- Specify the number of iterations and cases to simulate.

Select an item. Enables you to switch between the three views of the Simulation Generate node: Simulated Fields, Correlations, and Advanced Options.

Simulated Fields view

If the Simulation Generate node has been generated or updated from a Simulation Fitting node using historical data, in the Simulated Fields view you can view and edit the statistical distribution information for each field. The following information about each field is copied to the Types tab of the Simulation Generate node from the Simulation Fitting node:

- Measurement level
- Values
- Missing
- Check
- Role

If you do not have historical data, you can define fields and specify their distributions by selecting a storage type, and selecting a distribution type and entering the required parameters. Generating data in this way means that information about the measurement level of each field will not be available until the data are instantiated, for example, on the Types tab or in a Type node.

The Simulated Fields view contains several tools, which you can use to perform the following tasks:

- Add and remove fields.
- Change the order of fields on the display.
- Specify a storage type for each field.
- Specify a statistical distribution for each field.
- Specify parameter values for the statistical distribution of each field.

Simulated Fields. This table contains one empty row if the Simulation Generate node has been added to the stream canvas from the Sources palette. When this row is edited, a new empty row is added to the bottom of the table. If the Simulation Generate node has been created from a Simulation Fitting node, this table will contain one row for each field of the historical data. Extra rows can be added to the table by clicking the Add new field icon.

The Simulated Fields table is made up of the following columns:

- Field. Contains the names of the fields. The field names can be edited by typing in the cells.
- Storage. The cells in this column contain a drop-down list of storage types. Available storage types are String, Integer, Real, Time, Date, and Timestamp. The choice of storage type determines which distributions are available in the Distribution column. If the Simulation Generate node has been created from a Simulation Fitting node, the storage type is copied over from the Simulation Fitting node.
- Note: For fields with datetime storage types, you must specify the distribution parameters as integers. For example, to specify a mean date of 1 January 1970, use the integer 0. The signed integer represents the number of seconds since (or before) midnight on 1 January 1970.
- Status. Icons in the Status column indicate the fit status for each field.



No distribution has been specified for the field or one or more distribution parameter is missing. In order to run the simulation, you must specify a distribution for this field and enter valid values for the parameters.

	The field is set to the closest fitting distribution. Note: This icon can only ever be displayed if the Simulation Generate node is created from a Simulation Fitting node.
	The closest fitting distribution has been replaced with an alternative distribution from the Fit Details sub-dialog box. See the topic Fit Details for more information.
	The distribution has been manually specified or edited, and might include a parameter specified at more than one level.

- Locked. Locking a simulated field, by selecting the check box in the column with the lock icon, excludes the field from automatic updating by a linked Simulation Fitting node. This is most useful when you manually specify a distribution and want to ensure that it will not be affected by automatic distribution fitting when a linked Simulation Fitting node is executed.
- Distribution. The cells in this column contain a drop-down list of statistical distributions. The choice of storage type determines which distributions are available in this column for a given field. See the topic [Distributions](#) for more information.
Note: You cannot specify the Fixed distribution for every field. If you want every field in your generated data to be fixed, you can use a User Input node followed by a Balance node.
- Parameters. The distribution parameters associated with each fitted distribution are displayed in this column. Multiple values of a parameter are comma separated. Specifying multiple values for a parameter generates multiple iterations for the simulation. See the topic [Iterations](#) for more information. If parameters are missing, this is reflected in the icon displayed in the Status column. To specify values for the parameters, click this column in the row corresponding to the field of interest and choose Specify from the list. This opens the Specify Parameters sub-dialog box. See the topic [Specify Parameters](#) for more information. This column is disabled if Empirical is chosen in the Distribution column.
- Min, Max. In this column, for some distributions, you can specify a minimum value, a maximum value, or both for the simulated data. Simulated data that are smaller than the minimum value and larger than the maximum value will be rejected, even though they would be valid for the specified distribution. To specify minimum and maximum values, click this column in the row corresponding to the field of interest and choose Specify from the list. This opens the Specify Parameters sub-dialog box. See the topic [Specify Parameters](#) for more information. This column is disabled if Empirical is chosen in the Distribution column.

Use Closest Fit. Only enabled if the Simulation Generate node has been created automatically from a Simulation Fitting node using historical data, and a single row in the Simulated Fields table is selected. Replaces the information for the field in the selected row with the information of the best fitting distribution for the field. If the information in the selected row has been edited, pressing this button will reset the information back to the best fitting distribution determined from the Simulation Fitting node.

Fit Details. Only enabled if the Simulation Generate node has been created automatically from a Simulation Fitting node. Opens the Fit Details sub-dialog box. See the topic [Fit Details](#) for more information.

Several useful tasks can be carried out using the icons on the right of the Simulated Fields view. These icons are described in the following table.

Table 1. Icons on the Simulated Fields view

Icon	Tooltip	Description
	Edit distribution parameters	Only enabled when a single row in the Simulated Fields table is selected. Opens the Specify Parameters sub-dialog box for the selected row. See the topic Specify Parameters for more information.
	Add new field	Only enabled when a single row in the Simulated Fields table is selected. Adds a new empty row to the bottom of the Simulated Fields table.
	Create multiple copies	Only enabled when a single row in the Simulated Fields table is selected. Opens the Clone Field sub-dialog box. See the topic Clone Field for more information.
	Delete selected field	Deletes the selected row from the Simulated Fields table.
	Move to top	Only enabled if the selected row is not already the top row of the Simulated Fields table. Moves the selected row to the top of the Simulated Fields table. This action affects the order of the fields in the simulated data.
	Move up	Only enabled if the selected row is not the top row of the Simulated Fields table. Moves the selected row up one position in the Simulated Fields table. This action affects the order of the fields in the simulated data.
	Move down	Only enabled if the selected row is not the bottom row of the Simulated Fields table. Moves the selected row down one position in the Simulated Fields table. This action affects the order of the fields in the simulated data.
	Move to bottom	Only enabled if the selected row is not already the bottom row of the Simulated Fields table. Moves the selected row to the bottom of the Simulated Fields table. This action affects the order of the fields in the simulated data.

Do not clear Min and Max when refitting. When selected, the minimum and maximum values are not overwritten when the distributions are updated by executing a connected Simulation Fitting node.

Correlations view

Input fields to predictive models are often known to be correlated--for example, height and weight. Correlations between fields that will be simulated must be accounted for in order to ensure that the simulated values preserve those correlations.

If the Simulation Generate node has been generated or updated from a Simulation Fitting node using historical data, in the Correlations view you can view and edit the calculated correlations between pairs of fields. If you do not have historical data, you can specify the correlations manually based on your knowledge of how the fields are correlated.

Note: Before any data are generated, the correlation matrix is automatically checked to see if it is positive semi-definite, and can therefore be inverted. A matrix can be inverted if its columns are linearly independent. If the correlation matrix cannot be inverted, it will be automatically adjusted to make it invertible.

You can choose to display the correlations in a matrix or list format.

Correlations matrix. Displays the correlations between pairs of fields in a matrix. The field names are listed, in alphabetical order, down the left and along the top of the matrix. Only the cells below the diagonal can be edited; a value between -1.000 and 1.000, inclusive, must be entered. The cell above the diagonal is updated when focus is changed away from its mirrored cell below the diagonal; both cells then display the same value. The diagonal cells are always disabled and always have a correlation of 1.000. The default value for all other cells is 0.000. A value of 0.000 specifies that there is no correlation between the associated pair of fields. Only continuous and ordinal fields are included in the matrix. Nominal, categorical and flag fields, and fields that are assigned the Fixed distribution are not shown in the table.

Correlations list. Displays the correlations between pairs of fields in a table. Each row of the table shows the correlation between a pair of fields. Rows cannot be added or deleted. The columns with the headings Field 1 and Field 2 contain the field names, which cannot be edited. The Correlation column contains the correlations, which can be edited; a value between -1.000 and 1.000, inclusive, must be entered. The default value for all cells is 0.000. Only continuous and ordinal fields are included in the list. Nominal, categorical and flag fields, and fields that are assigned the Fixed distribution are not shown in the list.

Reset correlations. Opens the Reset Correlations dialog box. If historical data is available, you can choose one of three options:

- Fitted. Replaces the current correlations with those calculated using the historical data.
- Zeroes. Replaces the current correlations with zeroes.
- Cancel. Closes the dialog box. The correlations are unchanged.

If historical data is not available, but you have made changes to the correlations, you can choose to replace the current correlations with zeroes, or cancel.

Show As. Select Table to display the correlations as a matrix. Select List to display the correlations as a list.

Do not recalculate correlations when refitting. Select this option if you want to manually specify correlations and prevent them from being overwritten when automatically fitting distributions using a Simulation Fitting node and historical data.

Use fitted multiway contingency table for inputs with a categorical distribution. By default, all fields with a categorical distribution are included in a contingency table (or multiway contingency table, depending on the number of fields with a categorical distribution). The contingency table is constructed, like the correlations, when a Simulation Fitting node is executed. The contingency table cannot be viewed. When this option is selected, fields with a categorical distribution are simulated using the actual percentages from the contingency table. That is, any associations between nominal fields are recreated in the new, simulated data. When this option is cleared, fields with categorical distributions are simulated using the expected percentages from the contingency table. If you modify a field, the field is removed from the contingency table.

Advanced Options view

Number of cases to simulate. Displays the options for specifying the number of cases to be simulated, and how any iterations will be named.

- Maximum number of cases. This specifies the maximum number of cases of simulated data, and associated target values, to generate. The default value is 100,00, minimum value is 1000, and maximum value is 2,147,483,647.
- Iterations. This number is calculated automatically and cannot be edited. An iteration is created automatically each time a distribution parameter has multiple values specified.
- Total rows. Only enabled when the number of iterations is greater than 1. The number is calculated automatically, using the equation shown, and cannot be edited.
- Create iteration field. Only enabled when the number of iterations is greater than 1. When selected, the Name field is enabled. See the topic [Iterations](#) for more information.
- Name. Only enabled when the Create iteration field check box is selected, and the number of iterations is greater than 1. Edit the name of the iteration field by typing in this text field. See the topic [Iterations](#) for more information.

Random seed. Setting a random seed allows you to replicate your simulation.

- Replicate results. When selected, the Generate button and Random seed field are enabled.
- Random seed. Only enabled when the Replicate results check box is selected. In this field you can specify an integer to be used as the random seed. The default value is 629111597.
- Generate. Only enabled when the Replicate results check box is selected. Creates a pseudo-random integer between 1 and 999999999, inclusive, in the Random seed field.

Related information

- [Simulation Generate Node](#)

Clone Field

In the Clone Field dialog box you can specify how many copies of the selected field to create, and how each copy is named. It is useful to have multiple copies of fields when investigating compounded effects, for example interest or growth rates over a number of successive time periods.

The title bar of the dialog box contains the name of the selected field.

Number of copies to make. Contains the number of copies of the field to create. Click the arrows to select the number of copies to create. The minimum number of copies is 1 and the maximum is 512. The number of copies is initially set to 10.

Copy suffix character(s). Contains the characters that are added to the end of the field name for each copy. These characters separate the field name from the copy number. The suffix characters can be edited by typing in this field. This field can be left empty; in this case there will be no characters between the field name and copy number. The default character is an underscore.

Initial copy number. Contains the suffix number for the first copy. Click the arrows to select the initial copy number. The minimum initial copy number is 1 and the maximum is 1000. The default initial copy number is 1.

Copy number step. Contains the increment for the suffix numbers. Click the arrows to select the increment. The minimum increment is 1 and the maximum is 255. The increment is initially set to 1.

Fields. Contains a preview of the field names for the copies, which is updated if any of the fields of the Clone Field dialog box are edited. This text is generated automatically and cannot be edited.

OK. Generates all the copies as specified in the dialog box. The copies are added to the Simulated Fields table in the Simulation Generate node dialog box, directly below the row that contains the field that was copied.

Cancel. Closes the dialog box. Any changes that have been made are discarded.

Fit Details

The Fit Details dialog box is only available if the Simulation Generate node was created or updated by executing a Simulation Fitting node. It displays the results of automatic distribution fitting for the selected field. Distributions are ordered by goodness of fit, with the closest fitting distribution listed first. In this dialog box you can perform the following tasks:

- Examine the distributions fitted to the historical data.
- Select one of the fitted distributions.

Field. Contains the name of the selected field. This text cannot be edited.

Treat as (Measure). Displays the measurement type of the selected field. This is taken from the Simulated Fields table in the Simulation Generate node dialog box. The measurement type can be changed by clicking the arrow and selecting a measurement type from the drop-down list. There are three options: Continuous, Nominal, and Ordinal.

Distributions. The Distributions table shows all the distributions that are appropriate for the measurement type. The distributions that have been fitted to the historical data are ordered by goodness of fit, from best fitting to worst fitting. Goodness of fit is determined by the fit statistic chosen in the Simulation Fitting node. The distributions that were not fitted to the historical data are listed in the table in alphabetical order below the distributions that were fitted.

The Distribution table contains the following columns:

- **Use.** The selected radio button indicates which distribution is currently chosen for the field. You can override the closest fitting distribution by selecting the radio button for the desired distribution in the Use column. Selecting a radio button in the Use column also displays a plot of the distribution superimposed on a histogram (or bar chart) of the historical data for the selected field. Only one distribution can be selected at a time.
- **Distribution.** Contains the name of the distribution. This column cannot be edited.
- **Fit Statistics.** Contains the calculated fit statistics for the distribution. This column cannot be edited. The contents of the cell depends on the measurement type of the field:
 - **Continuous.** Contains the results of the Anderson-Darling and Kolmogorov-Smirnov tests. The p-values associated with the tests are also shown. The fit statistic chosen as the Goodness of fit criteria in the Simulation Fitting node is shown first and is used to order the distributions. The Anderson-Darling statistics are displayed as A=aval P=pval. The Kolmogorov-Smirnov statistics are displayed as K=kval P=pval. If a statistic cannot be calculated, a dot is displayed instead of a number.
 - **Nominal and Ordinal.** Contains the results of the chi-squared test. The p-value associated with the test is also shown. The statistics are displayed as Chi-Sq=val P=pval. If the distribution was not fitted, Not fitted is displayed. If the distribution cannot be mathematically fit, Cannot be fit is displayed.

Note: The cell is always empty for the Empirical distribution.

- **Parameters.** Contains the distribution parameters associated with each fitted distribution. The parameters are displayed as `parameter_name = parameter_value`, with the parameters separated by a single space. For the categorical distribution, the parameter names are the categories and the parameter values are the associated probabilities. If the distribution was not fitted to the historical data, the cell is empty. This column cannot be edited.

Histogram thumbnail. Shows a plot of the selected distribution superimposed on a histogram of the historical data of the selected field.

Distribution thumbnail. Shows an explanation and illustration of the selected distribution.

OK. Closes the dialog box and updates the values of the Measurement, Distribution, Parameters and Min, Max columns of the Simulated Fields table for the selected field with the information from the selected distribution. The icon in the Status column is also updated to reflect whether the selected distribution is the distribution with the closest fit to the data.

Cancel. Closes the dialog box. Any changes that have been made are discarded.

Specify Parameters

In the Specify Parameters dialog box you can manually specify the parameter values for the distribution of the selected field. You can also choose a different distribution for the selected field.

The Specify Parameters dialog box can be opened in three ways:

- Double click on a field name in the Simulated Fields table in the Simulation Generate node dialog box.
- Click the Parameters or Min, Max column of the Simulated Fields table, and choose Specify from the list.
- In the Simulated Fields table, select a row, and then click the Edit distribution parameters icon.

Field. Contains the name of the selected field. This text cannot be edited.

Distribution. Contains the distribution of the selected field. This is taken from the Simulated Fields table. The distribution can be changed by clicking the arrow and selecting a distribution from the drop-down list. The available distributions depend on the storage type of the selected field.

Sides. This option is only available for the when the dice distribution is selected in the Distribution field. Click the arrows to select the number of sides, or categories, to split the field in to. The minimum number of sides is two, and the maximum is 20. The number of sides is initially set to 6.

Distribution parameters. The Distribution parameters table contains one row for each parameter of the chosen distribution.

Note: Distribution uses a rate parameter, with a shape parameter $\alpha = k$ and an inverse scale parameter $\beta = 1/\theta$.

The table contains two columns:

- **Parameter.** Contains the names of the parameters. This column cannot be edited.
 - **Value(s).** Contains the values of the parameters. If the Simulation Generate node has been created or updated from a Simulation Fitting node, the cells in this column contain the parameter values that have been determined by fitting the distribution to the historical data. If the Simulation Generate node has been added to the stream canvas from the Source nodes palette, the cells in this column are empty. The values can be edited by typing in the cells. See the topic [Distributions](#) for more information about the parameters needed by each distribution, and acceptable parameter values.
- Multiple values for a parameter must be separated by commas. Specifying multiple values for a parameter will define multiple iterations of the simulation. You can only specify multiple values for one parameter.

Note: For fields with datetime storage types, you must specify the distribution parameters as integers. For example, to specify a mean date of 1 January 1970, use the integer 0.

Note: When the dice distribution is selected, the Distribution parameters table is slightly different. The table contains one row for each side (or category). The table contains a Value column and a Probability column. The Value column contains a label for each category. The default values for the labels are the integers 1-N, where N is the number of sides. The labels can be edited by typing in the cells. Any value can be entered into the cells. If you want to use a value that is not a number, the storage type of the data field must be changed to string if the storage type is not already set to string. The Probability column contains the probability of each category. The probabilities cannot be edited, and are calculated as 1/N.

Preview. Shows a sample plot of the distribution, based on the specified parameters. If two or more values have been specified for one parameter, sample plots for each value of the parameter are shown. If historical data is available for the selected field, the plot of the distribution is superimposed on a histogram of the historical data.

Optional Settings. Use these options to specify a minimum value, a maximum value, or both for the simulated data. Simulated data that are smaller than the minimum value and larger than the maximum value will be rejected, even though they would be valid for the specified distribution.

- **Specify minimum.** Select to enable the Reject values below field. The check box is disabled if the Empirical distribution is selected.
- **Reject values below.** Only enabled if Specify minimum is selected. Enter a minimum value for the simulated data. Any simulated values smaller than this value will be rejected.
- **Specify maximum.** Select to enable the Reject values above field. The check box is disabled if the Empirical distribution is selected.

- **Reject values above.** Only enabled if Specify maximum is selected. Enter a maximum value for the simulated data. Any simulated values larger than this value will be rejected.

OK. Closes the dialog box and updates the values of the Distribution, Parameters and Min, Max columns of the Simulated Fields table for the selected field. The icon in the Status column is also updated to reflect the selected distribution.

Cancel. Closes the dialog box. Any changes that have been made are discarded.

- [Iterations](#)

Iterations

If you have specified more than one value for a fixed field or a distribution parameter, an independent set of simulated cases - effectively, a separate simulation - is generated for each specified value. This allows you to investigate the effect of varying the field or parameter. Each set of simulated cases is referred to as an *iteration*. In the simulated data, the iterations are stacked up.

If the Create iteration field check box on the Advanced Options view of the Simulation Generate node dialog is selected, an iteration field is added to the simulated data as a nominal field with numeric storage. The name of this field can be edited by typing in the Name field on the Advanced Options view. This field contains a label which indicates to which iteration each simulated case belongs. The form of the labels depends on the type of iteration:

- **Iterating a fixed field.** The label is the name of the field, followed by an equals sign, followed by the value of the field for that iteration, that is
field_name = field_value
- **Iterating a distribution parameter.** The label is the name of the field, followed by a colon, followed by the name of the iterated parameter, followed by an equals sign, followed by the value of the parameter for that iteration, that is
field_name:parameter_name = parameter_value
- **Iterating a distribution parameter for a categorical or range distribution.** The label is the name of the field, followed by a colon, followed by "Iteration", followed by the iteration number, that is
field_name: Iteration iteration_number

Distributions

You can manually specify the probability distribution for any field by opening the Specify Parameters dialog box for that field, selecting the desired distribution from the Distribution list, and entering the distribution parameters in the Distribution parameters table. Following are some notes on particular distributions:

- **Categorical.** The categorical distribution describes an input field that has a fixed number of numeric values, referred to as categories. Each category has an associated probability such that the sum of the probabilities over all categories equals one.
Note: If you specify probabilities for the categories that do not sum to 1, you will receive a warning.
- **Negative Binomial - Failures.** Describes the distribution of the number of failures in a sequence of trials before a specified number of successes are observed. The parameter *Threshold* is the specified number of successes and the parameter *Probability* is the probability of success in any given trial.
- **Negative Binomial - Trials.** Describes the distribution of the number of trials that are required before a specified number of successes are observed. The parameter *Threshold* is the specified number of successes and the parameter *Probability* is the probability of success in any given trial.
- **Range.** This distribution consists of a set of intervals with a probability assigned to each interval such that the sum of the probabilities over all intervals equals 1. Values within a given interval are drawn from a uniform distribution defined on that interval. Intervals are specified by entering a minimum value, a maximum value and an associated probability.
For example, you believe that the cost of a raw material has a 40% chance of falling in the range of \$10 - \$15 per unit, and a 60% chance of falling in the range of \$15 - \$20 per unit. You would model the cost with a Range distribution consisting of the two intervals [10 - 15] and [15 - 20], setting the probability associated with the first interval to 0.4 and the probability associated with the second interval to 0.6. The intervals do not have to be contiguous and they can even be overlapping. For example, you might have specified the intervals \$10 - \$15 and \$20 - \$25 or \$10 - \$15 and \$13 - \$16.
- **Weibull.** The parameter *Location* is an optional location parameter, which specifies where the origin of the distribution is located.

The following table shows the distributions that are available for custom distribution fitting, and the acceptable values for the parameters. Some of these distributions are available for custom fitting to particular storage types, even though they are not fitted automatically to these storage types by the Simulation Fitting node.

Table 1. Distributions available for custom fitting

Distribution	Storage type supported for custom fitting	Parameters	Parameter limits	Notes
Bernoulli	Integer, real, datetime	Probability	$0 \leq \text{Probability} \leq 1$	
Beta	Integer, real, datetime	Shape 1 Shape 2 Minimum Maximum	≥ 0 ≥ 0 $< \text{Maximum}$ $> \text{Minimum}$	Minimum and maximum are optional.
Binomial	Integer, real, datetime	Number of trials (n) Probability Minimum Maximum	> 0 , integer $0 \leq \text{Probability} \leq 1$ $< \text{Maximum}$ $> \text{Minimum}$	Number of trials must be an integer. Minimum and maximum are optional.
Categorical	Integer, real, datetime, string	Category name (or label)	$0 \leq \text{Value} \leq 1$	Value is the probability of the category. The values must sum to 1, otherwise a warning is generated.
Dice	Integer, string	Sides	$2 \leq \text{Sides} \leq 20$	The probability of each category (side) is calculated as $1/N$, where N is the number of sides. The probabilities cannot be edited.
Empirical	Integer, real, datetime			You cannot edit the empirical distribution, or select it as a type. The Empirical distribution is only available when there is historical data.
Exponential	Integer, real, datetime	Scale Minimum Maximum	> 0 $< \text{Maximum}$ $> \text{Minimum}$	Minimum and maximum are optional.
Fixed	Integer, real, datetime, string	Value		You cannot specify the Fixed distribution for every field. If you want every field in your generated data to be fixed, you can use a User Input node followed by a Balance node.
Gamma	Integer, real, datetime	Shape Scale Minimum Maximum	≥ 0 ≥ 0 $< \text{Maximum}$ $> \text{Minimum}$	Minimum and maximum are optional. Distribution uses a rate parameter, with a shape parameter $\alpha = k$ and an inverse scale parameter $\beta = 1/\theta$.
Lognormal	Integer, real, datetime	Shape 1 Shape 2 Minimum Maximum	≥ 0 ≥ 0 $< \text{Maximum}$ $> \text{Minimum}$	Minimum and maximum are optional.
Negative Binomial - Failures	Integer, real, datetime	Threshold Probability Minimum Maximum	≥ 0 $0 \leq \text{Probability} \leq 1$ $< \text{Maximum}$ $> \text{Minimum}$	Minimum and maximum are optional.
Negative Binomial - Trials	Integer, real, datetime	Threshold Probability Minimum Maximum	≥ 0 $0 \leq \text{Probability} \leq 1$ $< \text{Maximum}$ $> \text{Minimum}$	Minimum and maximum are optional.
Normal	Integer, real, datetime	Mean Standard deviation Minimum Maximum	≥ 0 > 0 $< \text{Maximum}$ $> \text{Minimum}$	Minimum and maximum are optional.
Poisson	Integer, real, datetime	Mean Minimum Maximum	≥ 0 $< \text{Maximum}$ $> \text{Minimum}$	Minimum and maximum are optional.
Range	Integer, real, datetime	Begin(X) End(X) Probability(X)		X is the index of each bin. The probability values must sum to 1.
Triangular	Integer, real, datetime	Mode Minimum Maximum	$\text{Minimum} \leq \text{Value} \leq \text{Maximum}$ $< \text{Maximum}$ $> \text{Minimum}$	
Uniform	Integer, real, datetime	Minimum Maximum	$< \text{Maximum}$ $> \text{Minimum}$	

Distribution	Storage type supported for custom fitting	Parameters	Parameter limits	Notes
Weibull	Integer, real, datetime	Rate Scale Location Minimum Maximum	> 0 > 0 ≥ 0 < Maximum > Minimum	Location, maximum and minimum are optional.

Extension Import node

With the Extension Import node, you can run R or Python for Spark scripts to import data.

- [Extension Import node - Syntax tab](#)
- [Extension Import node - Console Output tab](#)
- [Filtering or renaming fields](#)
- [Viewing and setting information about types](#)

Extension Import node - Syntax tab

Select your type of syntax – R or Python for Spark. Then enter or paste your custom script for importing data. When your syntax is ready, you can click Run to execute the Extension Import node.

R example

```
# import R demo data cars to modeler
modelerData <- cars

# write the data model that matches the data
var1<-c(fieldName="speed",fieldLabel="",fieldStorage="integer",fieldMeasure="",fieldFormat="", fieldRole="")
var2<-c(fieldName="dist",fieldLabel="",fieldStorage="integer",fieldMeasure="",fieldFormat="", fieldRole="")
modelerDataModel<-data.frame(var1, var2)
```

Python for Spark example

```
import spss.pyspark.runtime
from pyspark.sql import SQLContext
from pyspark.sql.types import *

cxt = spss.pyspark.runtime.getContext()
if cxt.isComputeDataModelOnly():
    _schema = StructType([StructField("Age", LongType(), nullable=True), \
                          StructField("Sex", StringType(), nullable=True), \
                          StructField("BP", StringType(), nullable=True), \
                          StructField("Cholesterol", StringType(), nullable=True), \
                          StructField("Na", DoubleType(), nullable=True), \
                          StructField("K", DoubleType(), nullable=True), \
                          StructField("Drug", StringType(), nullable=True)])
    cxt.setSparkOutputSchema(_schema)
else:
    sqlContext = cxt.getSparkSQLContext()
    # the demo data is in modeler installation path
    df = sqlContext.read.option("inferSchema", "true").option("header",
    "true").csv("/opt/IBM/SPSS/ModelerServer/Cloud/demos/DRUG1n")
    cxt.setSparkOutputData(df)
    df.show()
    # print (df.dtypes[:])
```

Extension Import node - Console Output tab

The Console Output tab contains any output that is received when the R script or Python for Spark script on the Syntax tab runs (for example, if using an R script, it shows output received from the R console when the R script in the R Syntax field on the Syntax tab is executed). This output might include R or Python error messages or warnings that are produced when the R or Python script is executed. The output can be used, primarily, to debug the script. The Console Output tab also contains the script from the R Syntax or Python Syntax field.

Every time the Extension Import script is executed, the content of the Console Output tab is overwritten with the output received from the R console or Python for Spark. The output cannot be edited.

Filtering or renaming fields

You can rename or exclude fields at any point in a stream. For example, as a medical researcher, you may not be concerned about the potassium level (field-level data) of patients (record-level data); therefore, you can filter out the **K** (potassium) field. This can be done using a separate Filter node or using the Filter tab on a source or output node. The functionality is the same regardless of which node it is accessed from.

- From source nodes, such as Variable File, Fixed File, Statistics File, XML, or Extension Import, you can rename or filter fields as the data are read into IBM® SPSS® Modeler.
- Using a Filter node, you can rename or filter fields at any point in the stream.
- From Statistics Export, Statistics Transform, Statistics Model, and Statistics Output nodes, you can filter or rename fields to conform to IBM SPSS Statistics naming standards. See the topic [Renaming or Filtering Fields for IBM SPSS Statistics](#) for more information.
- You can use the Filter tab in any of the above nodes to define or edit multiple response sets. See the topic [Editing Multiple Response Sets](#) for more information.
- Finally, you can use a Filter node to map fields from one source node to another.

Viewing and setting information about types

From various source nodes as well as the Type node, you can specify field metadata and properties that are invaluable to modeling and other work in IBM® SPSS® Modeler. These properties include:

- Specifying a usage type, such as range, set, ordered set, or flag, for each field in your dataset.
- Setting options for handling missing values and system nulls.
- Setting the role of a field for modeling purposes.
- Specifying values for a field as well as options used to automatically read values from the dataset.
- Specifying field and value labels.

Select help specific to your situation from the list below.

Geospatial Source Node

You use the Geospatial source node to bring map or spatial data into your data mining session. You can import data in one of two ways:

- In a shape file (.shp)
- By connecting to an ESRI server that contains a hierarchical file system which includes map files.

Note: You can only connect to public map services.

Spatio-Temporal Prediction (STP) models can include map or spatial elements in their predictions. For more information, see [Spatio-Temporal Prediction modeling node](#)

- [Setting Options for the Geospatial Source Node](#)

Setting Options for the Geospatial Source Node

Datasource type You can import data from either a Shape file (.shp), or connect to a Map service.

If you are using a Shape file, either enter the file name and the filepath to it, or browse to select the file. The file must either be on the local directory or accessed from a mapped drive; you cannot access the file by using a uniform naming convention (UNC) path.

Note: Shape data requires both a .shp and a .dbf file. The two files must have the same name and be in the same folder. The .dbf file is automatically imported when you select the .shp file. In addition, there might be a .prj file that specifies the coordinate system for the shape data.

If you are using a Map service, enter the URL to the service and click Connect. After you connect to the service, the layers within that service are displayed in the bottom of the dialog box in a tree structure in the Available maps pane; expand the tree and select the layer that you require.

Note: You can only connect to public map services.

Automatic definition of geospatial data

By default, SPSS® Modeler automatically defines, when possible, any geospatial data fields in the source node with the correct metadata. Metadata might include the measurement level of the geospatial field (such as Point or Polygon) and the coordinate system that is used by the fields, including details such as the origin point (for example, latitude 0, longitude 0) and the units of measure. For more information about measurement levels, see [Geospatial measurement sublevels](#).

The .shp and .dbf files that make up the shape file contain a common identifier field that is used as a key. For example, the .shp file might contain countries, with the country name field used as an identifier, and the .dbf file might contain information about those countries with the name of the country that is also used as the identifier.

Note: If the coordinate system is not the same as the default SPSS Modeler coordinate system you may have to reproject the data to use the required coordinate system. For more information, see [Reprojection Node](#).

Related information

- [Geospatial Source Node](#)
 - [Reprojection Node](#)
-

JSON Source node

Use the JSON source node to import data from a JSON file into an SPSS® Modeler stream, using UTF-8 encoding. Data in a JSON file can be in form of *object*, *array*, or *value*. This JSON source node only supports reading in an *array* of objects, and the object can't be nested.

Example JSON data:

```
[  
  {  
    "After": 122762,  
    "Promotion": 1467,  
    "Cost": 23.99,  
    "Class": "Confection",  
    "Before": 11495  
  },  
  {  
    "After": 137097,  
    "Promotion": 1745,  
    "Cost": 79.29,  
    "Class": "Drink",  
    "Before": 123378  
  }  
]
```

When SPSS Modeler reads data from a JSON file, it performs the following translations.

Table 1. JSON data storage translation

JSON Value	SPSS Modeler Data Storage
string	String
number(int)	Integer
number(real)	Real
true	1(Integer)
false	0(Integer)
null	Missing values

The following options are available in the JSON source node dialog.

JSON data source. Select the JSON file to import.

JSON string format. Specify the format of the JSON string. Select Records if your JSON file is a collection of name and value pairs. The JSON source node imports names as field names in SPSS Modeler. Or select Values if your JSON data only uses values (no names).

Common Source Node Tabs

The following options can be specified for all source nodes by clicking the corresponding tab:

- **Data tab.** Used to change the default storage type.

- **Filter tab.** Used to eliminate or rename data fields. This tab offers the same functionality as the Filter node. See the topic [Setting filtering options](#) for more information.
 - **Types tab.** Used to set measurement levels. This tab offers the same functionality as the Type node.
 - **Annotations tab.** Used for all nodes, this tab offers options to rename nodes, supply a custom ToolTip, and store a lengthy annotation.
- [Setting Measurement Levels in the Source Node](#)**
- [Filtering Fields from the Source Node](#)**

Related information

- [Setting Field Storage and Formatting](#)
- [Setting Measurement Levels in the Source Node](#)

Setting Measurement Levels in the Source Node

Field properties can be specified in a source node or in a separate Type node. The functionality is similar in both nodes. The following properties are available:

- Field Double-click any field name to specify value and field labels for data in IBM® SPSS® Modeler. For example, field metadata imported from IBM SPSS Statistics can be viewed or modified here. Similarly, you can create new labels for fields and their values. The labels that you specify here are displayed throughout IBM SPSS Modeler depending on the selections you make in the stream properties dialog box.
- Measurement This is the measurement level, used to describe characteristics of the data in a given field. If all of the details of a field are known, it is called **fully instantiated**. For more information, see [Measurement levels](#).
Note: The measurement level of a field is different from its storage type, which indicates whether the data are stored as strings, integers, real numbers, dates, times, timestamps, or lists.
- Values This column enables you to specify options for reading data values from the data set, or use the Specify option to specify measurement levels and values in a separate dialog box. You can also choose to pass fields without reading their values. For more information, see [Data Values](#).
Note: You cannot amend the cell in this column if the corresponding Field entry contains a list.
- Missing Used to specify how missing values for the field will be handled. For more information, see [Defining Missing Values](#).
Note: You cannot amend the cell in this column if the corresponding Field entry contains a list.
- Check In this column, you can set options to ensure that field values conform to the specified values or ranges. For more information, see [Checking Type Values](#).
Note: You cannot amend the cell in this column if the corresponding Field entry contains a list.
- Role Used to tell modeling nodes whether fields will be Input (predictor fields) or Target (predicted fields) for a machine-learning process. Both and None are also available roles, along with Partition, which indicates a field used to partition records into separate samples for training, testing, and validation. The value Split specifies that separate models will be built for each possible value of the field. For more information, see [Setting the field role](#).

See the topic [Type Node](#) for more information.

- [When to instantiate at the source node](#)

Related information

- [Common Source Node Tabs](#)
- [Data Values](#)
- [Checking Type Values](#)
- [Setting the field role](#)

When to instantiate at the source node

There are two ways you can learn about the data storage and values of your fields. This *instantiation* can occur at either the source node, when you first bring data into IBM® SPSS® Modeler, or by inserting a Type node into the data stream.

Instantiating at the source node is useful when:

- The dataset is small.
- You plan to derive new fields using the Expression Builder (instantiating makes field values available from the Expression Builder).

Generally, if your dataset is not very large and you don't plan to add fields later in the stream, instantiating at the source node is the most convenient method.

Note: If you export data in a database export node, the data must be fully instantiated.

Filtering Fields from the Source Node

The Filter tab on a source node dialog box enables you to exclude fields from downstream operations based on your initial examination of the data. This is useful, for example, if there are duplicate fields in the data or if you are already familiar enough with the data to exclude irrelevant fields. Alternatively, you can add a separate Filter node later in the stream. The functionality is similar in both cases. See the topic [Setting filtering options](#) for more information.

Related information

- [Setting filtering options](#)

Record Operations Nodes

- [Overview of Record Operations](#)
- [Select Node](#)
- [Sample Node](#)
- [Balance Node](#)
- [Aggregate Node](#)
- [RFM Aggregate Node](#)
- [Sort Node](#)
- [Merge Node](#)
- [Append Node](#)
- [Distinct node](#)
- [Streaming Time Series node](#)
- [SMOTE node](#)
- [Extension Transform node](#)
- [Space-Time-Boxes Node](#)
- [Streaming TCM Node](#)
- [CPLEX Optimization node](#)

Overview of Record Operations

Record operations nodes are used to make changes to data at the record level. These operations are important during the **Data Understanding** and **Data Preparation** phases of data mining because they allow you to tailor the data to your particular business need.

For example, based on the results of the data audit conducted using the Data Audit node (Output palette), you might decide that you would like customer purchase records for the past three months to be merged. Using a Merge node, you can merge records based on the values of a key field, such as *Customer ID*. Or you might discover that a database containing information about Web site hits is unmanageable with over one million records. Using a Sample node, you can select a subset of data for use in modeling.

The Record Operations palette contains the following nodes:

	The Select node selects or discards a subset of records from the data stream based on a specific condition. For example, you might select the records that pertain to a particular sales region.
	The Sample node selects a subset of records. A variety of sample types are supported, including stratified, clustered, and nonrandom (structured) samples. Sampling can be useful to improve performance, and to select groups of related records or transactions for analysis.
	The Balance node corrects imbalances in a dataset, so it conforms to a specified condition. The balancing directive adjusts the proportion of records where a condition is true by the factor specified.
	The Aggregate node replaces a sequence of input records with summarized, aggregated output records.
	The Recency, Frequency, Monetary (RFM) Aggregate node enables you to take customers' historical transactional data, strip away any unused data, and combine all of their remaining transaction data into a single row that lists when they last dealt with you, how many transactions they have made, and the total monetary value of those transactions.
	The Sort node sorts records into ascending or descending order based on the values of one or more fields.
	The Merge node takes multiple input records and creates a single output record containing some or all of the input fields. It is

	useful for merging data from different sources, such as internal customer data and purchased demographic data.
	The Append node concatenates sets of records. It is useful for combining datasets with similar structures but different data.
	The Distinct node removes duplicate records, either by passing the first distinct record to the data stream or by discarding the first record and passing any duplicates to the data stream instead.
	The Streaming Time Series node builds and scores time series models in one step. You can use the node with data in either a local or distributed environment; in a distributed environment you can harness the power of IBM® SPSS® Analytic Server
	The Spectral Clustering© algorithm uses several eigenvectors to project data into a space with fewer dimensions. Then a k-means clustering algorithm is applied in the new space to separate the data into clusters. It's reasonably fast for small records with many fields, and computationally expensive for large data sets. The Spectral Clustering node in SPSS Modeler exposes the core features and commonly used parameters of the Spectral Clustering library. The node is implemented in Python.
	Space-Time-Boxes (STB) are an extension of Geohashed spatial locations. More specifically, an STB is an alphanumeric string that represents a regularly shaped region of space and time.
	The Streaming TCM node builds and scores temporal causal models in one step.
	The CPLEX Optimization node provides the ability to use complex mathematical (CPLEX) based optimization via an Optimization Programming Language (OPL) model file. This functionality was available in the IBM Analytical Decision Management product, which is no longer supported. But you can also use the CPLEX node in SPSS Modeler without requiring IBM Analytical Decision Management.

Many of the nodes in the Record Operations palette require you to use a CLEM expression. If you are familiar with CLEM, you can type an expression in the field. However, all expression fields provide a button that opens the CLEM Expression Builder, which helps you create such expressions automatically.

Figure 1. Expression Builder button



Select Node

You can use Select nodes to select or discard a subset of records from the data stream based on a specific condition, such as `BP = "HIGH"`.

Mode. Specifies whether records that meet the condition will be included or excluded from the data stream.

- **Include.** Select to include records that meet the selection condition.
- **Discard.** Select to exclude records that meet the selection condition.

Condition. Displays the selection condition that will be used to test each record, which you specify using a CLEM expression. Either enter an expression in the window or use the Expression Builder by clicking the calculator (Expression Builder) button to the right of the window.

If you choose to discard records based on a condition, such as the following:

```
(var1='value1' and var2='value2')
```

the Select node by default also discards records having null values for all selection fields. To avoid this, append the following condition to the original one:

```
and not(@NULL(var1) and @NULL(var2))
```

Select nodes are also used to choose a proportion of records. Typically, you would use a different node, the Sample node, for this operation. However, if the condition you want to specify is more complex than the parameters provided, you can create your own condition using the Select node. For example, you can create a condition such as:

```
BP = "HIGH" and random(10) <= 4
```

This will select approximately 40% of the records showing high blood pressure and pass those records downstream for further analysis.

Related information

- [Overview of Record Operations](#)
 - [Sample Node](#)
-

Sample Node

You can use Sample nodes to select a subset of records for analysis, or to specify a proportion of records to discard. A variety of sample types are supported, including stratified, clustered, and nonrandom (structured) samples. Sampling can be used for several reasons:

- To improve performance by estimating models on a subset of the data. Models estimated from a sample are often as accurate as those derived from the full dataset, and may be more so if the improved performance allows you to experiment with different methods you might not otherwise have attempted.
- To select groups of related records or transactions for analysis, such as selecting all the items in an online shopping cart (or market basket), or all the properties in a specific neighborhood.
- To identify units or cases for random inspection in the interest of quality assurance, fraud prevention, or security.

Note: If you simply want to partition your data into training and test samples for purposes of validation, a Partition node can be used instead. See the topic [Partition Node](#) for more information.

Types of Samples

Clustered samples. Sample groups or clusters rather than individual units. For example, suppose you have a data file with one record per student. If you cluster by school and the sample size is 50%, then 50% of schools will be chosen and all students from each selected school will be picked. Students in unselected schools will be rejected. On average, you would expect about 50% of students to be picked, but because schools vary in size, the percentage may not be exact. Similarly, you could cluster shopping cart items by transaction ID to make sure that all items from selected transactions are maintained. For an example that clusters properties by town, see the *complexsample_property.str* sample stream.

Stratified samples. Select samples independently within non-overlapping subgroups of the population, or strata. For example, you can ensure that men and women are sampled in equal proportions, or that every region or socioeconomic group within an urban population is represented. You can also specify a different sample size for each stratum (for example, if you think that one group has been under-represented in the original data). For an example that stratifies properties by county, see the *complexsample_property.str* sample stream.

Systematic or 1-in-n sampling. When selection at random is difficult to obtain, units can be sampled systematically (at a fixed interval) or sequentially.

Sampling weights. Sampling weights are automatically computed while drawing a complex sample and roughly correspond to the "frequency" that each sampled unit represents in the original data. Therefore, the sum of the weights over the sample should estimate the size of the original data.

Sampling Frame

A sampling frame defines the potential source of cases to be included in a sample or study. In some cases, it may be feasible to identify every single member of a population and include any one of them in a sample--for example, when sampling items that come off a production line. More often, you will not be able to access every possible case. For example, you cannot be sure who will vote in an election until after the election has happened. In this case, you might use the electoral register as your sampling frame, even though some registered people won't vote, and some people may vote despite not having been listed at the time you checked the register. Anybody not in the sampling frame has no prospect of being sampled. Whether your sampling frame is close enough in nature to the population you are trying to evaluate is a question that must be addressed for each real-life case.

- [Sample node options](#)
- [Cluster and Stratify Settings](#)
- [Sample Sizes for Strata](#)

Related information

- [Overview of Record Operations](#)
 - [Sample node options](#)
 - [Cluster and Stratify Settings](#)
 - [Sample Sizes for Strata](#)
-

Sample node options

You can choose the Simple or Complex method as appropriate for your requirements.

Simple sampling options

The Simple method allows you to select a random percentage of records, select contiguous records, or select every *n*th record.

Mode. Select whether to pass (include) or discard (exclude) records for the following modes:

- Include sample. Includes selected records in the data stream and discards all others. For example, if you set the mode to Include sample and set the 1-in-n option to 5, then every fifth record will be included, yielding a dataset that is roughly one-fifth the original size. This is the default mode when sampling data, and the only mode when using the complex method.
- Discard sample. Excludes selected records and includes all others. For example, if you set the mode to Discard sample and set the 1-in-n option to 5, then every fifth record will be discarded. This mode is only available with the simple method.

Sample. Select the method of sampling from the following options:

- First. Select to use contiguous data sampling. For example, if the maximum sample size is set to 10000, then the first 10,000 records will be selected.
- 1-in-n. Select to sample data by passing or discarding every *n*th record. For example, if *n* is set to 5, then every fifth record will be selected.
- Random %. Select to sample a random percentage of the data. For example, if you set the percentage to 20, then 20% of the data will either be passed to the data stream or discarded, depending on the mode selected. Use the field to specify a sampling percentage. You can also specify a seed value using the Set random seed control.

Use block level sampling (in-database only). This option is enabled only if you choose random percentage sampling when performing in-database mining on an Oracle or IBM Db2 database. In these circumstances, block-level sampling can be more efficient.

Note: You do not get an exact number of rows returned each time you run the same random sample settings. This is because each input record has a probability of $\frac{N}{100}$ of being included in the sample (where *N* is the Random % you specify in the node) and the probabilities are independent; therefore the results are not exactly *N*%.

Maximum sample size. Specifies the maximum number of records to include in the sample. This option is redundant and therefore disabled when First and Include are selected. Also note that when used in combination with the Random % option, this setting may prevent certain records from being selected. For example, if you have 10 million records in your dataset, and you select 50% of records with a maximum sample size of three million records, then 50% of the first six million records will be selected, and the remaining four million records have no chance of being selected. To avoid this limitation, select the Complex sampling method, and request a random sample of three million records without specifying a cluster or stratify variable.

Complex sampling options

Complex sample options allow for finer control of the sample, including clustered, stratified, and weighted samples along with other options.

Cluster and stratify. Allows you to specify cluster, stratify, and input weight fields if needed. See the topic [Cluster and Stratify Settings](#) for more information.

Sample type.

- Random. Selects clusters or records randomly within each strata.
- Systematic. Selects records at a fixed interval. This option works like the *1 in n* method, except the position of the first record changes depending on a random seed. The value of *n* is determined automatically based on the sample size or proportion.

Sample units. You can select proportions or counts as the basic sample units.

Sample size. You can specify the sample size in several ways:

- Fixed. Allows you to specify the overall size of the sample as a count or proportion.
- Custom. Allows you to specify the sample size for each subgroup or strata. This option is only available if a stratification field has been specified in the Cluster and Stratify sub dialog box.
- Variable. Allows the user to pick a field that defines the sample size for each subgroup or strata. This field should have the same value for each record within a particular stratum; for example, if the sample is stratified by county, then all records with county = Surrey must have the same value. The field must be numeric and its values must match the selected sample units. For proportions, values should be greater than 0 and less than 1; for counts, the minimum value is 1.

Minimum sample per stratum. Specifies a minimum number of records (or minimum number of clusters if a cluster field is specified).

Maximum sample per stratum. Specifies a maximum number of records or clusters. If you select this option without specifying a cluster or stratify field, a random or systematic sample of the specified size will be selected.

Set random seed. When sampling or partitioning records based on a random percentage, this option allows you to duplicate the same results in another session. By specifying the starting value used by the random number generator, you can ensure the same records are assigned each time the node is executed. Enter the desired seed value, or click the Generate button to automatically generate a random value. If this option is not selected, a different sample will be generated each time the node is executed.

Note: When using the Set random seed option with records read from a database, a Sort node may be required prior to sampling in order to ensure the same result each time the node is executed. This is because the random seed depends on the order of records, which is not guaranteed to stay the same in a relational database. See the topic [Sort Node](#) for more information.

Cluster and Stratify Settings

The Cluster and Stratify dialog box enables you to select cluster, stratification, and weight fields when drawing a complex sample.

Clusters. Specifies a categorical field used to cluster records. Records are sampled based on cluster membership, with some clusters included and others not. But if any record from a given cluster is included, all are included. For example, when analyzing product associations in shopping carts, you could cluster items by transaction ID to make sure that all items from selected transactions are maintained. Instead of sampling records—which would destroy information about what items are sold together—you can sample transactions to make sure that all records for selected transactions are preserved.

Stratify by. Specifies a categorical field used to stratify records so that samples are selected independently within non-overlapping subgroups of the population, or strata. If you select a 50% sample stratified by gender, for example, then two 50% samples will be taken, one for the men and one for the women. For example, strata may be socioeconomic groups, job categories, age groups, or ethnic groups, allowing you to ensure adequate sample sizes for subgroups of interest. If there are three times more women than men in the original dataset, this ratio will be preserved by sampling separately from each group. Multiple stratification fields can also be specified (for example, sampling product lines within regions or vice-versa).

Note: If you stratify by a field that has missing values (null or system missing values, empty strings, white space, and blank or user-defined missing values), then you cannot specify custom sample sizes for strata. If you want to use custom sample sizes when stratifying by a field with missing or blank values, then you need to fill them upstream.

Use input weight. Specifies a field used to weight records prior to sampling. For example, if the weight field has values ranging from 1 to 5, records weighted 5 are five times as likely to be selected. The values of this field will be overwritten by the final output weights generated by the node (see following paragraph).

New output weight. Specifies the name of the field where final weights are written if no input weight field is specified. (If an input weight field is specified, its values are replaced by the final weights as noted above, and no separate output weight field is created.) The output weight values indicate the number of records represented by each sampled record in the original data. The sum of the weight values gives an estimate of the sample size. For example, if a random 10% sample is taken, the output weight will be 10 for all records, indicating that each sampled record represents roughly ten records in the original data. In a stratified or weighted sample, the output weight values may vary based on the sample proportion for each stratum.

Comments

- Clustered sampling is useful if you cannot get a complete list of the population you want to sample, but can get complete lists for certain groups or clusters. It is also used when a random sample would produce a list of test subjects that it would be impractical to contact. For example, it would be easier to visit all farmers in one county than a selection of farmers scattered across every county in the nation.
- You can specify both cluster and stratify fields in order to sample clusters independently within each strata. For example, you could sample property values stratified by county, and cluster by town within each county. This will ensure that an independent sample of towns is drawn from within each county. Some towns will be included and others will not, but for each town that is included, all properties within the town are included.
- To select a random sample of units from within each cluster, you can string two Sample nodes together. For example, you could first sample townships stratified by county as described above. Then attach a second Sample node and select *town* as a stratify field, allowing you to sample a proportion of records from within each township.
- In cases where a combination of fields is required to uniquely identify clusters, a new field can be generated using a Derive node. For example, if multiple shops use the same numbering system for transactions, you could derive a new field that concatenates the shop and transaction IDs.

Related information

- [Sample Node](#)
- [Sample node options](#)
- [Sample Sizes for Strata](#)

Sample Sizes for Strata

When drawing a stratified sample, the default option is to sample the same proportion of records or clusters from each stratum. If one group outnumbers another by a factor of 3, for example, you typically want to preserve the same ratio in the sample. If this is not the case, however, you can specify the sample size separately for each stratum.

The Sample Sizes for Strata dialog box lists each value of the stratification field, allowing you to override the default for that stratum. If multiple stratification fields are selected, every possible combination of values is listed, allowing you to specify the size for each ethnic group within each city, for example, or each town within each county. Sizes are specified as proportions or counts, as determined by the current setting in the Sample node.

To Specify Sample Sizes for Strata

1. In the Sample node, select Complex, and select one or more stratification fields. See the topic [Cluster and Stratify Settings](#) for more information.
2. Select Custom, and select Specify Sizes.
3. In the Sample Sizes for Strata dialog box, click the Read Values button at lower left to populate the display. If necessary, you may need to instantiate values in an upstream source or Type node. See the topic [What is instantiation?](#) for more information.
4. Click in any row to override the default size for that stratum.

Notes on Sample Size

Custom sample sizes may be useful if different strata have different variances, for example, in order to make sample sizes proportional to the standard deviation. (If the cases within the stratum are more varied, you need to sample more of them to get a representative sample.) Or if a stratum is small, you may wish to use a higher sample proportion to ensure that a minimum number of observations is included.

Note: If you stratify by a field that has missing values (null or system missing values, empty strings, white space, and blank or user-defined missing values), then you cannot specify custom sample sizes for strata. If you want to use custom sample sizes when stratifying by a field with missing or blank values, then you need to fill them upstream.

Related information

- [Sample Node](#)
 - [Sample node options](#)
 - [Cluster and Stratify Settings](#)
-

Balance Node

You can use Balance nodes to correct imbalances in datasets so they conform to specified test criteria. For example, suppose that a dataset has only two values--*low* or *high*--and that 90% of the cases are *low* while only 10% of the cases are *high*. Many modeling techniques have trouble with such biased data because they will tend to learn only the *low* outcome and ignore the *high* one, since it is more rare. If the data are well balanced with approximately equal numbers of *low* and *high* outcomes, models will have a better chance of finding patterns that distinguish the two groups. In this case, a Balance node is useful for creating a balancing directive that reduces cases with a *low* outcome.

Balancing is carried out by duplicating and then discarding records based on the conditions you specify. Records for which no condition holds are always passed through. Because this process works by duplicating and/or discarding records, the original sequence of your data is lost in downstream operations. Be sure to derive any sequence-related values before adding a Balance node to the data stream.

Note: Balance nodes can be generated automatically from distribution charts and histograms. For example, you can balance your data to show equal proportions across all categories of a categorical field, as shown in a distribution plot.

Example. When building an RFM stream to identify recent customers who have positively responded to previous marketing campaigns, the marketing department of a sales company uses a Balance node to balance the differences between true and false responses in the data.

- [Setting Options for the Balance Node](#)

Related information

- [Overview of Record Operations](#)
 - [Setting Options for the Balance Node](#)
-

Setting Options for the Balance Node

Record balancing directives. Lists the current balancing directives. Each directive includes both a factor and a condition that tells the software to "increase the proportion of records by a factor specified where the condition is true." A factor lower than 1.0 means that the proportion of indicated records will be decreased. For example, if you want to decrease the number of records where drug Y is the treatment drug, you might create a balancing directive with a factor of 0.7 and a condition `Drug = "drugY"`. This directive means that the number of records where drug Y is the treatment drug will be reduced to 70% for all downstream operations.

Note: Balance factors for reduction may be specified to four decimal places. Factors set below 0.0001 will result in an error, since the results do not compute correctly.

- **Create conditions** by clicking the button to the right of the text field. This inserts an empty row for entering new conditions. To create a CLEM expression for the condition, click the Expression Builder button.
- **Delete directives** using the red delete button.
- **Sort directives** using the up and down arrow buttons.

Only balance training data. If a partition field is present in the stream, this option balances data in the training partition only. In particular, this may be useful if generating adjusted propensity scores, which require an unbalanced testing or validation partition. If no partition field is present in the stream (or if multiple partition fields are specified), then this option is ignored and all data are balanced.

Related information

- [Balance Node](#)
-

Aggregate Node

Aggregation is a data preparation task frequently used to reduce the size of a dataset. Before proceeding with aggregation, you should take time to clean the data, concentrating especially on missing values. Once you have aggregated, potentially useful information regarding missing values may be lost.

You can use an Aggregate node to replace a sequence of input records with summary, aggregated output records. For example, you might have a set of input sales records such as those shown in the following table.

Table 1. Sales record input example

Age	Sex	Region	Branch	Sales
23	M	S	8	4
45	M	S	16	4
37	M	S	8	5
30	M	S	5	7
44	M	N	4	9
25	M	N	2	11
29	F	S	16	6
41	F	N	4	8
23	F	N	6	2
45	F	N	4	5
33	F	N	6	10

You can aggregate these records with *Sex* and *Region* as key fields. Then choose to aggregate *Age* with the mode Mean and *Sales* with the mode Sum. Select Include record count in field in the Aggregate node dialog box and your aggregated output would be as shown in the following table.

Table 2. Aggregated record example

Age (mean)	Sex	Region	Sales (sum)	Record Count
35.5	F	N	25	4
29	F	S	6	1
34.5	M	N	20	2
33.75	M	S	20	4

From this you learn, for example, that the average age of the four female sales staff in the North region is 35.5, and the sum total of their sales was 25 units.

Note: Fields such as *Branch* are automatically discarded when no aggregate mode is specified.

- [Setting options for the Aggregate node](#)
- [Aggregate optimization settings](#)

Related information

- [Overview of Record Operations](#)
- [Setting options for the Aggregate node](#)

Setting options for the Aggregate node

On the Aggregate node you specify the following.

- One or more key fields to use as categories for the aggregation
- One or more aggregate fields for which to calculate the aggregate values
- One or more aggregation modes (types of aggregation) to output for each aggregate field

You can also specify the default aggregation modes to use for newly added fields, and use expressions (similar to formulae) to categorize aggregation.

Note that for added performance, aggregations operations may benefit from enabling parallel processing.

Key fields. Lists fields that can be used as categories for aggregation. Both continuous (numeric) and categorical fields can be used as keys. If you choose more than one key field, the values will be combined to produce a key value for aggregating records. One aggregated record will be generated for each unique key field. For example, if `Sex` and `Region` are your key fields, each unique combination of `M` and `F` with regions `N` and `S` (four unique combinations) will have an aggregated record. To add a key field, use the Field Chooser button to the right of the window.

The rest of the dialog box is split into two main areas - Basic Aggregates and Aggregate Expressions.

Basic Aggregates

Aggregate fields. Lists the fields for which values will be aggregated as well as the selected modes of aggregation. To add fields to this list, use the Field Chooser button on the right. The following aggregation modes are available.

Note: Some modes are not applicable to non-numeric fields (for example, Sum for a date/time field). Modes that cannot be used with a selected aggregate field are disabled.

- Sum. Select to return summed values for each key field combination. The sum is the total of the values, across all cases with nonmissing values.
- Mean. Select to return the mean values for each key field combination. The mean is a measure of central tendency, and is the arithmetic average (the sum divided by the number of cases).
- Min. Select to return minimum values for each key field combination.
- Max. Select to return maximum values for each key field combination.
- SDev. Select to return the standard deviation for each key field combination. The standard deviation is a measure of dispersion around the mean, and is equal to the square root of the variance measurement.
- Median. Select to return the median values for each key field combination. The median is a measure of central tendency that is not sensitive to outlying values (unlike the mean, which can be affected by a few extremely high or low values). Also known as the 50th percentile or 2nd quartile.
- Count. Select to return the count of non-null values for each key field combination.
- Variance. Select to return the variance values for each key field combination. The variance is a measure of dispersion around the mean, equal to the sum of squared deviations from the mean divided by one less than the number of cases.
- 1st Quartile. Select to return the 1st quartile (25th percentile) values for each key field combination.
- 3rd Quartile. Select to return the 3rd quartile (75th percentile) values for each key field combination.

Note: When running a stream containing an Aggregate node, the values returned for 1st and 3rd Quartiles when pushing back SQL to an Oracle database may differ from those returned in native mode.

Default mode. Specify the default aggregation mode to be used for newly added fields. If you frequently use the same aggregation, select one or more modes here and use the Apply to All button on the right to apply the selected modes to all fields listed above.

New field name extension. Select to add a suffix or prefix, such as `1` or `new`, to duplicate aggregated fields. For example, the result of a minimum values aggregation on the field `Age` will produce a field name called `Age_Min_1` if you have selected the suffix option and specified `1` as the extension. Note that aggregation extensions such as `_Min` or `Max_` are automatically added to the new field, indicating the type of aggregation performed. Select Suffix or Prefix to indicate your preferred extension style.

Include record count in field. Select to include an extra field in each output record called `Record_Count`, by default. This field indicates how many input records were aggregated to form each aggregate record. Create a custom name for this field by typing in the edit field.

Note: System null values are excluded when aggregates are computed, but they are included in the record count. Blank values, on the other hand, are included in both aggregation and record count. To exclude blank values, you can use a Filler node to replace blanks with null values. You can also remove blanks using a Select node.

Aggregate Expressions

Expressions are similar to formulas that are created from values, field names, operators, and functions. Unlike functions that work on a single record at a time, aggregate expressions operate on a group, set, or collection of records.

Note: You can only create aggregate expressions if the stream includes a database connection (by means of a Database source node).

New expressions are created as derived fields; to create an expression you use the *Database Aggregates* functions which are available from the Expression Builder.

For more information, see [The Expression Builder](#).

Note that there is a connection between the Key Fields and any aggregate expressions you create because the aggregate expressions are grouped by the key field.

Valid aggregate expressions are ones that evaluate to aggregate outcomes; a couple of examples of valid aggregate expressions, and the rules that govern them, are as follows:

- You can use scalar functions to combine multiple aggregation functions together to produce a single aggregation result. For example:

```
max(C01) - min(C01)
```

- An aggregation function can operate on the result of multiple scalar functions. For example:

```
sum (C01*C01)
```

Aggregate optimization settings

On the Optimization tab you specify the following.

Keys are contiguous. Select this option if you know that all records with the same key values are grouped together in the input (for example, if the input is sorted on the key fields). Doing so can improve performance.

Allow approximation for Median and Quartiles. The Order statistics (Median, 1st Quartile, and 3rd Quartile) are currently not supported when processing data in Analytic Server. If you are using Analytic Server you can select this check box to use an approximated value for these statistics instead which is calculated by binning the data and then computing an estimate for the statistic based on the distribution across the bins. By default, this option is unchecked.

Number of bins. Only available if you select the Allow approximation for Median and Quartiles check box. Select the number of bins to be used when estimating the statistic; the number of bins affects the Maximum error %. By default, the number of bins is 1000, which corresponds to a maximum error of 0.1 percent of the range.

Related information

- [Aggregate Node](#)
- [Setting options for the Aggregate node](#)

RFM Aggregate Node

The Recency, Frequency, Monetary (RFM) Aggregate node enables you to take customers' historical transactional data, strip away any unused data, and combine all of their remaining transaction data into a single row, using their unique customer ID as a key, that lists when they last dealt with you (recency), how many transactions they have made (frequency), and the total value of those transactions (monetary).

Before proceeding with any aggregation, you should take time to clean the data, concentrating especially on any missing values.

Once you have identified and transformed the data using the RFM Aggregate node, you may use an RFM Analysis node to carry out further analysis. See the topic [RFM Analysis Node](#) for more information.

Note that once the data file has been run through the RFM Aggregate node, it will not have any target values; therefore, before being able to use it as inputs for further predictive analysis with any modeling nodes such as C5.0 or CHAID, you will need to merge it with other customer data (for example, by matching the customer IDs). See the topic [Merge Node](#) for more information.

The RFM Aggregate and RFM Analysis nodes in IBM® SPSS® Modeler are set up to use independent binning; that is, they rank and bin data on each measure of recency, frequency, and monetary value, without regard to their values or the other two measures.

- [Setting Options for the RFM Aggregate Node](#)

Related information

- [Overview of Record Operations](#)
- [Setting Options for the RFM Aggregate Node](#)

Setting Options for the RFM Aggregate Node

The Settings tab of the RFM Aggregate node contains the following fields.

Calculate Recency relative to Specify the date from which the recency of transactions will be calculated. This may be either a Fixed date that you enter, or Today's date, as set by your system. Today's date is entered by default and is automatically updated when the node is executed.

Note: The display of the Fixed date may be different for different locales. For example, if the value **2007-8-10** is stored in your stream as **Fri Aug 10 00:00:00 CST 2007** it is a time and date in the time zone 'UTC+8'. However, it displays as **Thu Aug 9 12:00:00 EDT 2007** in the time zone 'UTC-8'.

IDs are contiguous If your data are presorted so that all records with the same ID appear together in the data stream, select this option to speed up processing. If your data are not presorted (or you are not sure), leave this option unselected, and the node will sort the data automatically.

ID Select the field to be used to identify the customer and their transactions. To display the fields from which you can select, use the Field Chooser button on the right.

Date Select the date field to be used to calculate recency against. To display the fields from which you can select, use the Field Chooser button on the right.

Note that this requires a field with a storage of date, or timestamp, in the appropriate format to use as input. For example, if you have a string field with values like **Jan 2007**, **Feb 2007**, and so on, you can convert this to a date field using a Filler node and the `to_date()` function. See the topic [Storage Conversion Using the Filler Node](#) for more information.

Value Select the field to be used to calculate the total monetary value of the customer's transactions. To display the fields from which you can select, use the Field Chooser button on the right. Note: This must be a numeric value.

New field name extension Select to append either a suffix or prefix, such as "12_month", to the newly generated recency, frequency, and monetary fields. Select Suffix or Prefix to indicate your preferred extension style. For example, this may be useful when examining several time periods.

Discard records with value below If required, you can specify a minimum value below which any transaction details are not used when calculating the RFM totals. The units of value relate to the Value field selected.

Include only recent transactions If you are analyzing a large database, you can specify that only the latest records are used. You can chose to use the data recorded either after a certain date or within a recent period:

- Transaction date after Specify the transaction date after which records will be included in your analysis.
- Transaction within the last Specify the number and type of periods (days, weeks, months, or years) back from the Calculate recency relative to date after which records will be included in your analysis.

Save date of second most recent transaction If you want to know the date of the second most recent transaction for each customer, select this box. In addition, you can then select the Save date of third most recent transaction box as well. For example, this can help you identify customers who may have carried out many transactions some considerable time ago, but only one recent transaction.

Related information

- [RFM Aggregate Node](#)

Sort Node

You can use Sort nodes to sort records into ascending or descending order based on the values of one or more fields. For example, Sort nodes are frequently used to view and select records with the most common data values. Typically, you would first aggregate the data using the Aggregate node and then use the Sort node to sort the aggregated data into descending order of record counts. Displaying these results in a table will allow you to explore the data and to make decisions, such as selecting the records of the top 10 best customers.

The Settings tab of the Sort node contains the following fields.

Sort by. All fields selected to use as sort keys are displayed in a table. A key field works best for sorting when it is numeric.

- Add fields to this list using the Field Chooser button on the right.
- Select an order by clicking the Ascending or Descending arrow in the table's *Order* column.
- Delete fields using the red delete button.
- Sort directives using the up and down arrow buttons.

Default sort order. Select either Ascending or Descending to use as the default sort order when new fields are added above.

Note: The Sort node is not applied if there is a Distinct node down the model stream. For information about the Distinct node, see [Distinct node](#).

- [Sort Optimization Settings](#)

Related information

- [Overview of Record Operations](#)
 - [Sort Optimization Settings](#)
-

Sort Optimization Settings

If you are working with data you know are already sorted by some key fields, you can specify which fields are already sorted, allowing the system to sort the rest of the data more efficiently. For example, you want to sort by *Age* (descending) and *Drug* (ascending) but know your data are already sorted by *Age* (descending).

Data is presorted. Specifies whether the data are already sorted by one or more fields.

Specify existing sort order. Specify the fields that are already sorted. Using the Select Fields dialog box, add fields to the list. In the *Order* column, specify whether each field is sorted in ascending or descending order. If you are specifying multiple fields, make sure that you list them in the correct sorting order. Use the arrows to the right of the list to arrange the fields in the correct order. If you make a mistake in specifying the correct existing sort order, an error will appear when you run the stream, displaying the record number where the sorting is inconsistent with what you specified.

Note: Sorting speed may benefit from enabling parallel processing.

Related information

- [Sort Node](#)
-

Merge Node

The function of a Merge node is to take multiple input records and create a single output record containing all or some of the input fields. This is a useful operation when you want to merge data from different sources, such as internal customer data and purchased demographic data. You can merge data in the following ways.

- Merge by Order concatenates corresponding records from all sources in the order of input until the smallest data source is exhausted. It is important if using this option that you have sorted your data using a Sort node.
- Merge using a Key field, such as *Customer ID*, to specify how to match records from one data source with records from the other(s). Several types of joins are possible, including inner join, full outer join, partial outer join, and anti-join. See the topic [Types of Joins](#) for more information.
- Merge by Condition means that you can specify a condition to be satisfied for the merge to take place. You can specify the condition directly in the node, or build the condition using the Expression Builder.
- Merge by Ranked Condition is a left sided outer join in which you specify a condition to be satisfied for the merge to take place and a ranking expression which sorts into order from low to high. Most often used to merge geospatial data, you can specify the condition directly in the node, or build the condition using the Expression Builder.

- [Types of Joins](#)
- [Specifying a Merge Method and Keys](#)
- [Selecting Data for Partial Joins](#)
- [Specifying Conditions for a Merge](#)
- [Specifying Ranked Conditions for a Merge](#)
- [Filtering Fields from the Merge Node](#)
- [Setting Input Order and Tagging](#)
- [Merge Optimization Settings](#)

Related information

- [Sort Node](#)
- [Setting Input Order and Tagging](#)
- [Specifying a Merge Method and Keys](#)
- [Filtering Fields from the Merge Node](#)

Types of Joins

When using a key field for data merging, it is useful to spend some time thinking about which records will be excluded and which will be included. There are a variety of joins, which are discussed in detail below.

The two basic types of joins are referred to as inner and outer joins. These methods are frequently used to merge tables from related datasets based on common values of a key field, such as *Customer ID*. Inner joins allow for clean merging and an output dataset that includes only complete records. Outer joins also include complete records from the merged data, but they also allow you to include unique data from one or more input tables.

The types of joins allowed are described in greater detail below.

	An inner join includes only records in which a value for the key field is common to all input tables. That is, unmatched records will not be included in the output dataset.
	A full outer join includes all records, both matching and nonmatching, from the input tables. Left and right outer joins are referred to as partial outer joins and are described below.
	A partial outer join includes all records matched using the key field as well as unmatched records from specified tables. (Or, to put it another way, all records from some tables and only matching records from others.) Tables (such as A and B shown here) can be selected for inclusion in the outer join using the Select button on the Merge tab. Partial joins are also called left or right outer joins when only two tables are being merged. Since IBM® SPSS® Modeler allows the merging of more than two tables, we refer to this as a partial outer join.
	An anti-join includes only unmatched records for the first input table (Table A shown here). This type of join is the opposite of an inner join and does not include complete records in the output dataset.

For example, if you have information about farms in one dataset and farm-related insurance claims in another, you can match the records from the first source to the second source using the Merge options.

To determine if a customer in your farm sample has filed an insurance claim, use the inner join option to return a list showing where all IDs match from the two samples.

Figure 1. Sample output for an inner join merge

	id	name	region	farmsize	rainfall	landquality	farmincome	maincrop	claimtype	claimvalue
1	id604	name604	southwest	1860.000	103.0...	3.000	625251.000	potatoes	decomm...	281082.0...
2	id605	name605	north	1700.000	46.000	8.000	621148.000	wheat	decomm...	122006.0...
3	id620	name620	north	880.000	74.000	6.000	426988.000	rapeseed	arable_de	118885.0...

Using the full outer join option returns both matching and nonmatching records from the input tables. The system-missing value (\$null\$) will be used for any incomplete values.

Figure 2. Sample output for a full outer join merge

	id	name	region	farmsize	rainfall	landquality	farmincome	maincrop	claimtype	claimvalue
1	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$	decomm...	74703.1C
2	id602	name602	north	1780.000	42.000	9.000	734118.000	maize	\$null\$	\$null\$
3	id604	name604	southwest	1860.000	103.0...	3.000	625251.000	potatoes	decomm...	281082.0
4	id605	name605	north	1700.000	46.000	8.000	621148.000	wheat	decomm...	122006.0
5	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$	arable_de	122135.0

A partial outer join includes all records matched using the key field as well as unmatched records from specified tables. The table displays all of the records matched from the ID field as well as the records matched from the first dataset.

Figure 3. Sample output for a partial outer join merge

	id	claimtype	claimvalue	name	region	farmsize	rainfall	landquality	farmincome	maincrop
1	id602	\$null\$	\$null\$	name602	north	1780.000	42.000	9.000	734118.000	maize
2	id604	decomm...	281082.0...	name604	southwest	1860.000	103.0...	3.000	625251.000	potatoes
3	id605	decomm...	122006.0...	name605	north	1700.000	46.000	8.000	621148.000	wheat
4	id607	\$null\$	\$null\$	name607	southeast	1820.000	29.000	6.000	211605.000	maize
5	id608	\$null\$	\$null\$	name608	southeast	1640.000	108.0...	7.000	1167040.0...	maize
6	id609	\$null\$	\$null\$	name609	southwest	1600.000	101.0...	5.000	758755.000	wheat
7	id615	\$null\$	\$null\$	name615	midlands	920.000	86.000	6.000	442554.000	potatoes
8	id618	\$null\$	\$null\$	name618	southeast	1180.000	98.000	3.000	368646.000	maize

If you are using the anti-join option, the table returns only unmatched records for the first input table.

Figure 4. Sample output for an anti-join merge

	id	name	region	farmsize	rainfall	landquality	farmincome	maincrop
1	id602	name602	north	1780.000	42.000	9.000	734118.000	maize
2	id607	name607	southeast	1820.000	29.000	6.000	211605.000	maize
3	id608	name608	southeast	1640.000	108.0...	7.000	1167040.0...	maize
4	id609	name609	southwest	1600.000	101.0...	5.000	756755.000	wheat
5	id615	name615	midlands	920.000	66.000	6.000	442554.000	potatoes
6	id618	name618	southeast	1180.000	98.000	3.000	368646.000	maize
7	id619	name619	north	840.000	64.000	8.000	457552.000	potatoes

Specifying a Merge Method and Keys

The Merge tab of the Merge node contains the following fields.

Merge Method Select the method to be used for merging records. Selecting either Keys or Condition activates the bottom half of the dialog box.

- Order Merges records by order such that the *n*th record from each input is merged to produce the *n*th output record. When any record runs out of a matching input record, no more output records are produced. This means that the number of records that are created is the number of records in the smallest dataset. It is important if using this option that you have sorted your data using a Sort node.
- Keys Uses a key field, such as *Transaction ID*, to merge records with the same value in the key field. This is equivalent to a database "equi-join." If a key value occurs more than once, all possible combinations are returned. For example, if records with the same key field value A contain differing values B, C, and D in other fields, the merged fields produce a separate record for each combination of A with value B, A with value C, and A with value D.

Note: Null values are not considered identical in the merge-by-key method and will not join.

- Condition Use this option to specify a condition for the merge. For more information, see [Specifying Conditions for a Merge](#).
- Ranked condition Use this option to specify whether each row pairing in the primary and all secondary data sets are to be merged; use the ranking expression to sort any multiple matches into order from low to high. For more information, see [Specifying Ranked Conditions for a Merge](#).

Possible keys Lists only those fields with exactly matching field names in all input data sources. Select a field from this list and use the arrow button to add it as a key field used for merging records. More than one key field can be used. You can rename non-matching input fields by using a Filter node, or the Filter tab of a source node.

Keys for merge Lists all fields that are used to merge records from all input data sources based on values of the key fields. To remove a key from the list, select one and use the arrow button to return it to the Possible Keys list. When more than one key field is selected, the option below is enabled.

Combine duplicate key fields When more than one key field is selected above, this option ensures that there is only one output field of that name. This option is enabled by default except in the case when streams have been imported from earlier versions of IBM® SPSS® Modeler. When this option is disabled, duplicate key fields must be renamed or excluded by using the Filter tab in the Merge node dialog box.

Include only matching records (inner join) Select to merge only complete records.

Include matching and non-matching records (full outer join) Select to perform a "full outer join." This means that if values for the key field are not present in all input tables, the incomplete records are still retained. The undefined value (\$null\$) is added to the key field and included in the output record.

Include matching and selected non-matching records (partial outer join) Select to perform a "partial outer join" of the tables you select in a subdialog box. Click Select to specify tables for which incomplete records will be retained in the merge.

Include records in the first dataset not matching any others (anti-join) Select to perform a type of "anti-join," where only nonmatching records from the first dataset are passed downstream. You can specify the order of input datasets using arrows on the Inputs tab. This type of join does not include complete records in the output dataset. For more information, see [Types of Joins](#).

Related information

- [Merge Node](#)
- [Sort Node](#)
- [Filtering Fields from the Merge Node](#)

Selecting Data for Partial Joins

For a partial outer join, you must select the table(s) for which incomplete records will be retained. For example, you may want to retain all records from a Customer table while retaining only matched records from the Mortgage Loan table.

To select datasets for partial joins:

1. Click the Merge tab of the Merge node.

2. Select Include matching and selected non-matching records (partial outer join).
3. Click the Select button to open a dialog box where you can specify datasets for the partial join.

Outer Join column. In the *Outer Join* column, select datasets to include in their entirety. For a partial join, overlapping records will be retained as well as incomplete records for datasets selected here. See the topic [Types of Joins](#) for more information.

Related information

- [Merge Node](#)

Specifying Conditions for a Merge

By setting the merge method to Condition, you can specify one or more conditions to be satisfied for the merge to take place.

You can either enter the conditions directly into the Condition field, or build them with the aid of the Expression Builder by clicking the calculator symbol to the right of the field.

Add tags to duplicate field names to avoid merge conflicts If two or more of the datasets to be merged contain the same field names, select this check box to add a different prefix tag to the start of the field column headers. For example, if there are two fields called *Name* the result of the merge would contain *1_Name* and *2_Name*. If the tag has been renamed in the data source, the new name is used instead of the numbered prefix tag. If you do not select this check box, and there are duplicate names in the data, a warning is displayed to the right of the check box.

Related information

- [Merge Node](#)

Specifying Ranked Conditions for a Merge

A Ranked Condition merge can be considered as a left sided outer join merge by condition; the left side of the merge is the primary data set where each record is an event. For example, in a model that is used to find patterns in crime data, each record in the primary data set would be a crime and its associated information (location, type, and so on). In this example, the right side might contain the relevant geospatial data sets.

The merge uses both a merge condition and a ranking expression. The merge condition can use a geospatial function such as *within* or *close_to*. During the merge, all of the fields in the right side data sets are added to the left side data set but multiple matches result in a list field. For example:

- Left side: Crime data
- Right side: Counties data set and roads data set
- Merge conditions: Crime data *within* counties and *close_to* roads, along with a definition of what counts as *close_to*.

In this example, if a crime occurred within the required *close_to* distance of three roads (and the number of matches to be returned is set to at least three), then all three roads are returned as a list item.

By setting the merge method to Ranked condition, you can specify one or more conditions to be satisfied for the merge to take place.

Primary dataset Select the primary data set for the merge; the fields from all other data sets are added to the data set you select. This can be considered as the left side of an outer join merge.

When you select a primary data set, all the other input data sets that are connected to the Merge node are automatically listed in the Merges table.

Add tags to duplicate field names to avoid merge conflicts If two or more of the data sets to be merged contain the same field names, select this check box to add a different prefix tag to the start of the field column headers. For example, if there are two fields that are called *Name* the result of the merge would contain *1_Name* and *2_Name*. If the tag has been renamed in the data source, the new name is used instead of the numbered prefix tag. If you do not select this check box, and there are duplicate names in the data, a warning is displayed to the right of the check box.

Merges

Dataset

Shows the name of the secondary data sets that are connected as inputs to the Merge node. By default, where there is more than one secondary data set, they are listed in the order in which they were connected to the Merge node.

Merge Condition

Enter the unique conditions for merging each of the data sets in the table with the primary data set. You can either type the conditions directly into the cell, or build them with the aid of the Expression Builder by clicking the calculator symbol to the right of the cell. For example, you might use geospatial predicates to create a merge condition that places crime data from one data set within the county data of another data set. The default merge condition depends on the geospatial measurement level, as shown in the list below.

- Point, LineString, MultiPoint, MultiLineString - default condition of *close_to*.
- Polygon, MultiPolygon - default condition of *within*.

For more information about these levels, see [Geospatial measurement sublevels](#).

If a data set contains multiple geospatial fields of different types, the default condition that is used depends on the first measurement level that is found in the data, in the following descending order.

- Point
- LineString
- Polygon

Note: Defaults are only available when there is a geospatial data field in the secondary database.

Ranking Expression

Specify an expression to rank the merging of the data sets; this expression is used to sort multiple matches into an order that is based on the ranking criteria. You can either type the conditions directly into the cell, or build them with the aid of the Expression Builder by clicking the calculator symbol to the right of the cell.

Default ranking expressions of distance and area are provided in the Expression Builder and both rank low to high, meaning that, for example, the top match for distance is the smallest value. An example of ranking by distance is when the primary data set contains crimes and their associated location and each other data set contains objects with locations; in this case the distance between the crimes and the objects can be used as a ranking criteria. The default ranking expression depends on the geospatial measurement level, as shown in the list below.

- Point, LineString, MultiPoint, MultiLineString - the default expression is *distance*.
- Polygon, MultiPolygon - the default expression is *area*.

Note: Defaults are only available when there is a geospatial data field in the secondary database.

Number of Matches

Specify the number of matches that are returned, based on the condition and ranking expressions. The default number of matches depend on the geospatial measurement level in the secondary data set, as shown in the list below; however, you can double-click in the cell to enter your own value, up to a maximum of 100.

- Point, LineString, MultiPoint, MultiLineString - default value of 3.
- Polygon, MultiPolygon - default value of 1.
- Data set that contains no geospatial fields - default value of 1.

As an example, if you set up a merge that is based on a Merge Condition of *close_to* and a Ranking Expression of *distance*, the top three (closest) matches from the secondary data sets to each record in the primary data set are returned as the values in the resultant list field.

Related information

- [Merge Node](#)
-

Filtering Fields from the Merge Node

Merge nodes include a convenient way of filtering or renaming duplicate fields as a result of merging multiple data sources. Click the Filter tab in the dialog box to select filtering options.

The options presented here are nearly identical to those for the Filter node. There are, however, additional options not discussed here that are available on the Filter menu. See the topic [Filtering or renaming fields](#) for more information.

Field. Displays the input fields from currently connected data sources.

Tag. Lists the tag name (or number) associated with the data source link. Click the Inputs tab to alter active links to this Merge node.

Source node. Displays the source node whose data is being merged.

Connected node. Displays the node name for the node that is connected to the Merge node. Frequently, complex data mining requires several merge or append operations that may include the same source node. The connected node name provides a way of differentiating these.

Filter. Displays the current connections between input and output field. Active connections show an unbroken arrow. Connections with a red X indicate filtered fields.

Field. Lists the output fields after merging or appending. Duplicate fields are displayed in red. Click in the Filter field above to disable duplicate fields.

View current fields. Select to view information on fields selected to be used as key fields.

View unused field settings. Select to view information on fields that are not currently in use.

Related information

- [Merge Node](#)
-

Setting Input Order and Tagging

Using the Inputs tab in the Merge and Append node dialog boxes, you can specify the order of input data sources and make any changes to the tag name for each source.

Tags and order of input datasets. Select to merge or append only complete records.

- **Tag.** Lists current tag names for each input data source. Tag names, or **tags**, are a way of uniquely identifying the data links for the merge or append operation. For example, imagine water from various pipes that is combined at one point and flows through a single pipe. Data in IBM® SPSS® Modeler flows similarly, and the merging point is often a complex interaction between the various data sources. Tags provide a way of managing the inputs ("pipes") to a Merge or Append node so that if the node is saved or disconnected, the links remain and are easily identifiable.

When you connect additional data sources to a Merge or Append node, default tags are automatically created using numbers to represent the order in which you connected the nodes. This order is unrelated to the order of fields in the input or output datasets. You can change the default tag by entering a new name in the *Tag* column.

- **Source Node.** Displays the source node whose data is being combined.
- **Connected Node.** Displays the node name for the node that is connected to the Merge or Append node. Frequently, complex data mining requires several merge operations that may include the same source node. The connected node name provides a way of differentiating these.
- **Fields.** Lists the number of fields in each data source.

View current tags. Select to view tags that are actively being used by the Merge or Append node. In other words, current tags identify links to the node that have data flowing through. Using the pipe metaphor, current tags are analogous to pipes with existing water flow.

View unused tag settings. Select to view tags, or links, that were previously used to connect to the Merge or Append node but are not currently connected with a data source. This is analogous to empty pipes still intact within a plumbing system. You can choose to connect these "pipes" to a new source or remove them. To remove unused tags from the node, click Clear. This clears all unused tags at once.

Related information

- [Merge Node](#)
 - [Filtering Fields from the Merge Node](#)
-

Merge Optimization Settings

The system provides two options that can help you merge data more efficiently in certain situations. These options allow you to optimize merging when one input dataset is significantly larger than the other datasets or when your data are already sorted by all or some of the key fields that you are using for the merge.

Note: Optimizations from this tab only apply to IBM® SPSS® Modeler native node execution only; that is, when the Merge node does not pushback to SQL. Optimization settings have no effect on SQL generation.

One input dataset is relatively large. Select to indicate that one of the input datasets is much larger than the others. The system will cache the smaller datasets in memory and then perform the merge by processing the large dataset without caching or sorting it. You will commonly use this type of join with data designed using a star-schema or similar design, where there is a large central table of shared data (for example, in transactional data). If you select this option, click *Select* to specify the large dataset. Note that you can select only *one* large dataset. The following table summarizes which joins can be optimized using this method.

Table 1. Summary of join optimizations

Type of Join	Can be optimized for a large input dataset?
Inner	Yes

Type of Join	Can be optimized for a large input dataset?
Partial	Yes, if there are no incomplete records in the large dataset.
Full	No
Anti-join	Yes, if the large dataset is the first input.

All inputs are already sorted by key field(s). Select to indicate that the input data are already sorted by one or more of the key fields that you are using for the merge. Make sure *all* your input datasets are sorted.

Specify existing sort order. Specify the fields that are already sorted. Using the Select Fields dialog box, add fields to the list. You can select from only the key fields that are being used for the merge (specified in the Merge tab). In the Order column, specify whether each field is sorted in ascending or descending order. If you are specifying multiple fields, make sure that you list them in the correct sorting order. Use the arrows to the right of the list to arrange the fields in the correct order. If you make a mistake in specifying the correct existing sort order, an error will appear when you run the stream, displaying the record number where the sorting is inconsistent with what you specified.

Depending on the case sensitivity of the collation method used by the database, optimization may not function correctly where one or more inputs are sorted by the database. For example, if you have two inputs where one is case sensitive and the other is case insensitive, the results of sorting could be different. Merge optimization causes records to be processed using their sorted order. As a result, if inputs are sorted using different collation methods, the Merge node reports an error and displays the record number where sorting is inconsistent. When all inputs are from one source, or are sorted using mutually inclusive collations, records can be merged successfully.

Note: Merging speed may benefit from enabling parallel processing.

Related information

- [Merge Node](#)
-

Append Node

You can use Append nodes to concatenate sets of records. Unlike Merge nodes, which join records from different sources together, Append nodes read and pass downstream all of the records from one source until there are no more. Then the records from the next source are read using the same data structure (number of records, number of fields, and so on) as the first, or primary, input. When the primary source has more fields than another input source, the system null string (\$null\$) will be used for any incomplete values.

Append nodes are useful for combining datasets with similar structures but different data. For example, you might have transaction data stored in different files for different time periods, such as a sales data file for March and a separate one for April. Assuming that they have the same structure (the same fields in the same order), the Append node will join them together into one large file, which you can then analyze.

Note: In order to append files, the field measurement levels must be similar. For example, a *Nominal* field cannot be appended with a field whose measurement level is *Continuous*.

- [Setting Append Options](#)

Related information

- [Sort Node](#)
 - [Merge Node](#)
-

Setting Append Options

Match fields by. Select a method to use when matching fields to append.

- **Position.** Select to append datasets based on the position of fields in the main data source. When using this method, your data should be sorted to ensure proper appending.
- **Name.** Select to append datasets based on the name of fields in the input datasets. Also select Match case to enable case sensitivity when matching field names.

Output Field. Lists the source nodes that are connected to the Append node. The first node on the list is the primary input source. You can sort the fields in the display by clicking on the column heading. This sorting does not actually reorder the fields in the dataset.

Include fields from. Select Main dataset only to produce output fields based on the fields in the main dataset. The main dataset is the first input, specified on the Inputs tab. Select All datasets to produce output fields for all fields in all datasets regardless of whether there is a matching field across all input datasets.

Tag records by including source dataset in field. Select to add an additional field to the output file whose values indicate the source dataset for each record. Specify a name in the text field. The default field name is *Input*.

Related information

- [Append Node](#)
-

Distinct node

Duplicate records in a data set must be removed before data mining can begin. For example, in a marketing database, individuals may appear multiple times with different address or company information. You can use the Distinct node to find or remove duplicate records in your data, or to create a single, composite record from a group of duplicate records.

To use the Distinct node, you must first define a set of key fields that determine when two records are considered to be duplicates.

If you do not pick all your fields as key fields, then two "duplicate" records may not be truly identical because they can still differ in the values of the remaining fields. In this case, you can also define a sort order that is applied within each group of duplicate records. This sort order gives you fine control over which record is treated as the first within a group. Otherwise, all duplicates are considered to be interchangeable and any record might be selected. The incoming order of the records is not taken into account, so it does not help to use an upstream Sort node (see "Sorting records within the distinct node" below).

Mode. Specify whether to create a composite record, or to either include or exclude (discard) the first record.

- **Create a composite record for each group.** Provides a way for you to aggregate non-numeric fields. Selecting this option makes the Composite tab available where you specify how to create the composite records. See [Distinct Composite Settings](#) for more information.
- **Include only the first record in each group.** Selects the first record from each group of duplicate records and discards the rest. The *first* record is determined by the sort order defined below, and not by the incoming order of the records.
- **Discard only the first record in each group.** Discards the first record from each group of duplicate records and selects the remainder instead. The *first* record is determined by the sort order defined below, and not by the incoming order of the records. This option is useful for *finding* duplicates in your data so that you can examine them later in the stream.

Key fields for grouping. Lists the field or fields used to determine whether records are identical. You can:

- Add fields to this list using the field picker button on the right.
- Delete fields from the list by using the red X (remove) button.

Within groups, sort records by. Lists the fields used to determine how records are sorted within each group of duplicates, and whether they are sorted in ascending or descending order. You can:

- Add fields to this list using the field picker button on the right.
- Delete fields from the list by using the red X (remove) button.
- Move fields using the up or down buttons, if you are sorting by more than one field.

You must specify a sort order if you have chosen to include or exclude the first record in each group, and it matters to you which record is treated as the first.

Default sort order. Specify whether, by default, records are sorted in Ascending or Descending order of the sort key values.

Sorting records within the Distinct node

If the order of records within a group of duplicates is important to you, then you must specify the order using the Within groups, sort records by option in the Distinct node. Do not rely on an upstream Sort node. Remember that the incoming order of the records is not taken into account -- only the order specified within the node.

If you do not specify any sort fields (or you specify insufficient sort fields), then the records within each group of duplicates will be unordered (or incompletely ordered) and the results may be unpredictable.

For example, assume we have a very large set of log records pertaining to a number of machines. The log contains data such as the following:

Table 1. Machine log data

Timestamp	Machine	Temperature
17:00:22	Machine A	31
13:11:30	Machine B	26
16:49:59	Machine A	30
18:06:30	Machine X	32
16:17:33	Machine A	29

Timestamp	Machine	Temperature
19:59:04	Machine C	35
19:20:55	Machine Y	34
15:36:14	Machine X	28
12:30:41	Machine Y	25
14:45:49	Machine C	27
19:42:00	Machine B	34
20:51:09	Machine Y	36
19:07:23	Machine X	33

To reduce the number of records down to the latest record for each machine, use **Machine** as the key field and use **Timestamp** as the sort field (in descending order). The input order does not affect the result because the sort selection specifies which of the many rows for a given Machine is to be returned, and the final data output would be as follows.

Table 2. Sorted machine log data

Timestamp	Machine	Temperature
17:00:22	Machine A	31
19:42:00	Machine B	34
19:59:04	Machine C	35
19:07:23	Machine X	33
20:51:09	Machine Y	36

- [Distinct Optimization Settings](#)
- [Distinct Composite Settings](#)

Related information

- [Overview of Record Operations](#)
-

Distinct Optimization Settings

If the data on which you are working has only a small number of records, or has already been sorted, you can optimize the way in which it is handled to enable IBM® SPSS® Modeler to process the data more efficiently.

Note: If you either select Input dataset has a low number of distinct keys, or use SQL generation for the node, any row within the distinct key value can be returned; to control which row is returned within a distinct key you need to specify the sort order by using the Within groups, sort records by fields on the Settings tab. The optimization options do not affect the results output by the Distinct node as long as you have specified a sort order on the Settings tab.

Input dataset has a low number of distinct keys. Select this option if you have a small number of records, or a small number of unique values of the key field(s), or both. Doing so can improve performance.

Input dataset is already ordered by grouping fields and sorting fields on the Settings tab. Only select this option if your data is already sorted by all of the fields listed under Within groups, sort records by on the Settings tab, and if the ascending or descending sort order of the data is the same. Doing so can improve performance.

Disable SQL generation. Select this option to disable SQL generation for the node.

Related information

- [Overview of Record Operations](#)
-

Distinct Composite Settings

If the data on which you are working has multiple records, for example for the same person, you can optimize the way in which the data is handled by creating a single composite, or aggregate, record to process.

Note: This tab is only available when you select Create a composite record for each group on the Settings tab.

Setting options for the Composite tab

Field. This column shows all fields, except key fields in the data model, in their natural sort order. If the node is not connected, no fields are shown. To sort the rows alphabetically by field name, click the column header. You can select more than one row using either Shift-click or Ctrl-click. In addition, if you right-click a field, a menu is displayed from which you can choose to select all rows, sort the rows by ascending or descending field name or value, select fields by either measure or storage type, or select a value to automatically add the same Fill with values based on entry to every selected row.

Fill with values based on. Select the value type to be used for the composite record for the Field. The available options depend on the field type.

- For numeric range fields you can choose from:
 - First record in group
 - Last record in group
 - Total
 - Mean
 - Minimum
 - Maximum
 - Custom
- For time or date fields you can choose from:
 - First record in group
 - Last record in group
 - Earliest
 - Most Recent
 - Custom
- For string or typeless fields you can choose from:
 - First record in group
 - Last record in group
 - First alphanumeric
 - Last alphanumeric
 - Custom

In each case, you can use the Custom option to exercise more control over which value is used to fill the composite record. See [Distinct Composite - Custom Tab](#) for more information.

Include record count in field. Select this option to include an extra field in each output record, called **Record_Count** by default. This field indicates how many input records were aggregated to form each aggregate record. To create a custom name for this field, type your entry in the edit field.

- [Distinct Composite - Custom Tab](#)

Distinct Composite - Custom Tab

The Custom Fill dialog box gives you more control over which value is used to complete the new composite record. Note that must instantiate your data first before using this option if only customizing a single Field row on the Composite tab.

Note: This dialog box is only available when you select the Custom value in the Fill with values based on column on the Composite tab. Depending on the field type, you can choose from one of the following options.

- **Select by frequency.** Choose a value based on the frequency with which it occurs in the data record.
Note: Not available for Fields with a with a type of Continuous, Typeless, or Date/Time.
 - **Use.** Select from either Most or Least Frequent.
 - **Ties.** If there are two or more records with the same frequency of occurrence, specify how to select the required record. You can choose from one of four options: Use First, Use Last, Use Lowest, or Use Highest.
- **Includes value (T/F).** Select this to convert a field to a flag which identifies if any of the records in a group has a specified value. You can then select the Value from the list of those for the selected field.
Note: Not available if you select more than one Field row on the Composite tab
- **First match in list.** Select this to prioritize which value to give to the composite record. You can then select one of the Items from the list of those for the selected field.
Note: Not available if you select more than one Field row on the Composite tab
- **Concatenate values.** Select this to retain all the values in a group by concatenating them into a string. You must specify a delimiter to be used between each value.
Note: This is the only option available if you select one or more Field rows with a type of Continuous, Typeless, or Date/Time.
- **Use delimiter.** You can choose to use a Space or Comma as a delimiter value in the concatenated string. Alternatively, in the Other field, you can enter your own delimiter value character.
Note: Only available if you select the Concatenate values option.

Related information

- [Overview of Record Operations](#)
-

Streaming Time Series node

You use the Streaming Time Series node to build and score time series models in one step. A separate time series model is built for each target field, however model nuggets are not added to the generated models palette and the model information cannot be browsed.

Methods for modeling time series data require a uniform interval between each measurement, with any missing values indicated by empty rows. If your data does not already meet this requirement, you will have to transform values as needed.

Other points to note in connection with Time Series nodes are:

- Fields must be numeric.
- Date fields cannot be used as inputs.
- Partitions are ignored.

The Streaming Time Series node estimates exponential smoothing, univariate Autoregressive Integrated Moving Average (ARIMA), and multivariate ARIMA (or transfer function) models for time series and produces forecasts based on the time series data. Also available is an Expert Modeler, which attempts to automatically identify and estimate the best-fitting ARIMA or exponential smoothing model for one or more target fields.

For more information about time series modeling, see [Time series data](#).

The Streaming Time Series node is supported for use in a streaming deployment environment, through IBM® SPSS® Modeler Solution Publisher, using the IBM SPSS Collaboration and Deployment Services Scoring Service.

- [Streaming Time Series node - field options](#)
 - [Streaming Time Series node - data specification options](#)
 - [Streaming Time Series node - build options](#)
 - [Streaming Time Series node - model options](#)
-

Streaming Time Series node - field options

Use predefined roles This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign targets, predictors and other roles, select this option.

Note: If you have partitioned your data, the partitions are taken into account if you select Use predefined roles, but not if you select Use custom field assignments.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Targets Select one or more of the fields as your target for the prediction.

Candidate inputs Select one or more fields as inputs for the prediction.

Events and Interventions Use this area to designate certain input fields as event or intervention fields. This designation identifies a field as containing time series data that can be affected by events (predictable recurring situations; for example, sales promotions) or interventions (one-time incidents; for example, power outage or employee strike).

Streaming Time Series node - data specification options

The Data Specifications tab is where you set all the options for the data to be included in your model. As long as you specify both a Date/time field and Time interval, you can click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains several different panes on which you set the customizations that are specific to your model.

- [Streaming Time Series node - observations](#)
- [Streaming Time Series node - time interval for analysis](#)

- [Streaming Time Series node - aggregation and distribution options](#)
 - [Streaming Time Series node - missing value options](#)
 - [Streaming Time Series node - estimation period](#)
-

Streaming Time Series node - observations

Use the settings in this pane to specify the fields that define the observations.

Observations that are specified by a date/time field

You can specify that the observations are defined by a date, time, or timestamp field. In addition to the field that defines the observations, select the appropriate time interval that describes the observations. Depending on the specified time interval, you can also specify other settings, such as the interval between observations (increment) or the number of days per week. The following considerations apply to the time interval:

- Use the value Irregular when the observations are irregularly spaced in time, such as the time at which a sales order is processed. When Irregular is selected, you must specify the time interval that is used for the analysis, from the Time Interval settings on the Data Specifications tab.
- When the observations represent a date and time and the time interval is hours, minutes, or seconds, then use Hours per day, Minutes per day, or Seconds per day. When the observations represent a time (duration) without reference to a date and the time interval is hours, minutes, or seconds, then use Hours (non-periodic), Minutes (non-periodic), or Seconds (non-periodic).
- Based on the selected time interval, the procedure can detect missing observations. Detecting missing observations is necessary since the procedure assumes that all observations are equally spaced in time and that no observations are missing. For example, if the time interval is Days and the date 2015-10-27 is followed by 2015-10-29, then an observation is missing for 2015-10-28. Values are imputed for any missing observations; use the Missing Value Handling area of the Data Specifications tab to specify settings for handling missing values.
- The specified time interval allows the procedure to detect multiple observations in the same time interval that need to be aggregated together and to align observations on an interval boundary, such as the first of the month, to ensure that the observations are equally spaced. For example, if the time interval is Months, then multiple dates in the same month are aggregated together. This type of aggregation is referred to as *grouping*. By default, observations are summed when grouped. You can specify a different method for grouping, such as the mean of the observations, from the Aggregation and Distribution settings on the Data Specifications tab.
- For some time intervals, the additional settings can define breaks in the normal equally spaced intervals. For example, if the time interval is Days, but only weekdays are valid, you can specify that there are five days in a week, and the week begins on Monday.

Observations that are defined as periods or cyclic periods

Observations can be defined by one or more integer fields that represent periods or repeating cycles of periods, up to an arbitrary number of cycle levels. With this structure, you can describe series of observations that don't fit one of the standard time intervals. For example, a fiscal year with only 10 months can be described with a cycle field that represents years and a period field that represents months, where the length of one cycle is 10.

Fields that specify cyclic periods define a hierarchy of periodic levels, where the lowest level is defined by the Period field. The next highest level is specified by a cycle field whose level is 1, followed by a cycle field whose level is 2, and so on. Field values for each level, except the highest, must be periodic with respect to the next highest level. Values for the highest level cannot be periodic. For example, in the case of the 10-month fiscal year, months are periodic within years and years are not periodic.

- The length of a cycle at a particular level is the periodicity of the next lowest level. For the fiscal year example, there is only one cycle level and the cycle length is 10 since the next lowest level represents months and there are 10 months in the specified fiscal year.
- Specify the starting value for any periodic field that does not start from 1. This setting is necessary for detecting missing values. For example, if a periodic field starts from 2 but the starting value is specified as 1, then the procedure assumes that there is a missing value for the first period in each cycle of that field.

Streaming Time Series node - time interval for analysis

The time interval that you use for analysis can differ from the time interval of the observations. For example, if the time interval of the observations is Days, you might choose Months for the time interval for analysis. The data is then aggregated from daily to monthly data before the model is built. You can also choose to distribute the data from a longer to a shorter time interval. For example, if the observations are quarterly then you can distribute the data from quarterly to monthly data.

Use the settings in this pane to specify the time interval for the analysis. The method by which the data is aggregated or distributed is specified from the Aggregation and Distribution settings on the Data Specifications tab.

The available choices for the time interval at which the analysis is done depend on how the observations are defined and the time interval of those observations. In particular, when the observations are defined by cyclic periods, only aggregation is supported. In that case, the time interval of the analysis must be greater than or equal to the time interval of the observations.

Streaming Time Series node - aggregation and distribution options

Use the settings in this pane to specify settings for aggregating or distributing the input data with respect to the time intervals of the observations.

Aggregation functions

When the time interval that is used for the analysis is longer than the time interval of the observations, the input data are aggregated. For example, aggregation is done when the time interval of the observations is Days and the time interval for analysis is Months. The following aggregation functions are available: mean, sum, mode, min, or max.

Distribution functions

When the time interval that is used for the analysis is shorter than the time interval of the observations, the input data are distributed. For example, distribution is done when the time interval of the observations is Quarters and the time interval for analysis is Months. The following distribution functions are available: mean or sum.

Grouping functions

Grouping is applied when observations are defined by date/times and multiple observations occur in the same time interval. For example, if the time interval of the observations is Months, then multiple dates in the same month are grouped and associated with the month in which they occur. The following grouping functions are available: mean, sum, mode, min, or max. Grouping is always done when the observations are defined by date/times and the time interval of the observations is specified as Irregular.

Note: Although grouping is a form of aggregation, it is done before any handling of missing values whereas formal aggregation is done after any missing values are handled. When the time interval of the observations is specified as Irregular, aggregation is done only with the grouping function.

Aggregate cross-day observations to previous day

Specifies whether observations with times that cross a day boundary are aggregated to the values for the previous day. For example, for hourly observations with an eight-hour day that starts at 20:00, this setting specifies whether observations between 00:00 and 04:00 are included in the aggregated results for the previous day. This setting applies only if the time interval of the observations is Hours per day, Minutes per day or Seconds per day and the time interval for analysis is Days.

Custom settings for specified fields

You can specify aggregation, distribution, and grouping functions on a field by field basis. These settings override the default settings for the aggregation, distribution, and grouping functions.

Streaming Time Series node - missing value options

Use the settings in this pane to specify how any missing values in the input data are to be replaced with an imputed value. The following replacement methods are available:

Linear interpolation

Replaces missing values by using a linear interpolation. The last valid value before the missing value and the first valid value after the missing value are used for the interpolation. If the first or last observation in the series has a missing value, then the two nearest non-missing values at the beginning or end of the series are used.

Replaces missing values by using a linear interpolation.

- For non-seasonal data, the last valid value before the missing value and the first valid value after the missing value are used for the interpolation. If missing values are at the beginning or the end of a time series, then a linear extrapolation method is used based on the two nearest valid values.
- For seasonal data, a missing value is linearly interpolated using the last valid value of the same period before the missing value and the first valid value of the same period after the missing value. If one of the two values of the same period can't be found for the missing value, then the data will be regarded as non-seasonal data and linear interpolation of non-seasonal data is used to impute the missing value.

Series mean

Replaces missing values with the mean for the entire series.

Mean of nearby points

Replaces missing values with the mean of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the mean.

Median of nearby points

Replaces missing values with the median of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the median.

Linear trend

This option uses all non-missing observations in the series to fit a simple linear regression model, which is then used to impute the missing values.

Other settings:

Lowest data quality score (%)

Computes data quality measures for the time variable and for input data corresponding to each time series. If the data quality score is lower than this threshold, the corresponding time series will be discarded.

Streaming Time Series node - estimation period

In the Estimation Period pane, you can specify the range of records to be used in model estimation. By default, the estimation period starts at the time of the earliest observation and ends at the time of the latest observation across all series.

By start and end times

You can specify both the start and end of the estimation period or you can specify just the start or just the end. If you omit the start or the end of the estimation period, the default value is used.

- If the observations are defined by a date/time field, enter values for start and end in the same format that is used for the date/time field.
- For observations that are defined by cyclic periods, specify a value for each of the cyclic periods fields. Each field is displayed in a separate column.

By latest or earliest time intervals

Defines the estimation period as a specified number of time intervals that start at the earliest time interval or end at the latest time interval in the data, with an optional offset. In this context, the time interval refers to the time interval of the analysis. For example, assume that the observations are monthly but the time interval of the analysis is quarters. Specifying Latest and a value of 24 for the Number of time intervals means the latest 24 quarters.

Optionally, you can exclude a specified number of time intervals. For example, specifying the latest 24 time intervals and 1 for the number to exclude means that the estimation period consists of the 24 intervals that precede the last one.

Streaming Time Series node - build options

The Build Options tab is where you set all the options for building your model. You can, of course, just click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains two different panes on which you set the customizations that are specific to your model.

- [Streaming Time Series node - general build options](#)
-

Streaming Time Series node - general build options

The options available on this pane depend on which of the following three settings you choose from the Method list:

- Expert Modeler. Choose this option to use the Expert Modeler, which automatically finds the best-fitting model for each dependent series.
- Exponential Smoothing. Use this option to specify a custom exponential smoothing model.
- ARIMA. Use this option to specify a custom ARIMA model.

Expert Modeler

Under Model Type, select the type of models you want to build:

- All models. The Expert Modeler considers both ARIMA and exponential smoothing models.
- Exponential smoothing models only. The Expert Modeler considers only exponential smoothing models.
- ARIMA models only. The Expert Modeler considers only ARIMA models.

Expert Modeler considers seasonal models. This option is only enabled if a periodicity is defined for the active dataset. When this option is selected, the Expert Modeler considers both seasonal and nonseasonal models. If this option is not selected, the Expert Modeler considers only nonseasonal models.

Expert Modeler considers sophisticated exponential smoothing models. When this option is selected, the Expert Modeler searches a total of 13 exponential smoothing models (7 of them existed in the original Time Series node, and 6 of them were added in version 18.1). If this option is not selected, the Expert Modeler only searches the original 7 exponential smoothing models.

Under Outliers, select from the following options

Detect outliers automatically. By default, automatic detection of outliers is not performed. Select this option to perform automatic detection of outliers, then select the desired outlier types.

For more information, see [Outliers](#).

Input fields must have a measurement level of *Flag*, *Nominal*, or *Ordinal* and must be numeric (for example, 1/0, not True/False, for a flag field), before they are included in this list.

For more information, see [Pulses and steps](#).

The Expert Modeler considers only simple regression and not arbitrary transfer functions for inputs that are identified as event or intervention fields on the Fields tab.

Exponential Smoothing

Model Type. Exponential smoothing models are classified as either seasonal or nonseasonal.¹ Seasonal models are only available if the periodicity defined by using the Time Intervals pane on the Data Specifications tab is seasonal. The seasonal periodicities are: cyclic periods, years, quarters, months, days per week, hours per day, minutes per day, and seconds per day. The following model types are available:

- Simple. This model is appropriate for a series in which there is no trend or seasonality. Its only relevant smoothing parameter is level. Simple exponential smoothing is most similar to an ARIMA with zero orders of autoregression, one order of differencing, one order of moving average, and no constant.
- Holt's linear trend. This model is appropriate for a series in which there is a linear trend and no seasonality. Its relevant smoothing parameters are level and trend, and, in this model, they are not constrained by each other's values. Holt's model is more general than Brown's model but may take longer to compute estimates for large series. Holt's exponential smoothing is most similar to an ARIMA with zero orders of autoregression, two orders of differencing, and two orders of moving average.
- Damped trend. This model is appropriate for a series in which there is a linear trend that is dying out and no seasonality. Its relevant smoothing parameters are level, trend, and damping trend. Damped exponential smoothing is most similar to an ARIMA with one order of autoregression, one order of differencing, and two orders of moving average.
- Multiplicative trend. This model is appropriate for a series in which there is a trend that changes with the magnitude of the series and no seasonality. Its relevant smoothing parameters are level and trend. Multiplicative trend exponential smoothing is not similar to any ARIMA model.
- Brown's linear trend. This model is appropriate for a series in which there is a linear trend and no seasonality. Its relevant smoothing parameters are level and trend, but, in this model, they are assumed to be equal. Brown's model is therefore a special case of Holt's model. Brown's exponential smoothing is most similar to an ARIMA with zero orders of autoregression, two orders of differencing, and two orders of moving average, with the coefficient for the second order of moving average equal to one half of the coefficient for the first order squared.
- Simple seasonal. This model is appropriate for a series in which there is no trend and a seasonal effect that is constant over time. Its relevant smoothing parameters are level and season. Seasonal exponential smoothing is most similar to an ARIMA with zero orders of autoregression; one order of differencing; one order of seasonal differencing; and orders 1, p , and $p+1$ of moving average, where p is the number of periods in a seasonal interval. For monthly data, $p = 12$.
- Winters' additive. This model is appropriate for a series in which there is a linear trend and a seasonal effect that is constant over time. Its relevant smoothing parameters are level, trend, and season. Winters' additive exponential smoothing is most similar to an ARIMA with zero orders of autoregression; one order of differencing; one order of seasonal differencing; and $p+1$ orders of moving average, where p is the number of periods in a seasonal interval. For monthly data, $p=12$.
- Damped trend with additive seasonal. This model is appropriate for a series in which there is a linear trend that is dying out and a seasonal effect that is constant over time. Its relevant smoothing parameters are level, trend, damping trend, and season. Damped trend and additive seasonal exponential smoothing is not similar to any ARIMA model.
- Multiplicative trend with additive seasonal. This model is appropriate for a series in which there is a trend that changes with the magnitude of the series and a seasonal effect that is constant over time. Its relevant smoothing parameters are level, trend, and season. Multiplicative trend and additive seasonal exponential smoothing is not similar to any ARIMA model.
- Multiplicative seasonal. This model is appropriate for a series in which there is no trend and a seasonal effect that changes with the magnitude of the series. Its relevant smoothing parameters are level and season. Multiplicative seasonal exponential smoothing is not similar to any ARIMA model.
- Winters' multiplicative. This model is appropriate for a series in which there is a linear trend and a seasonal effect that changes with the magnitude of the series. Its relevant smoothing parameters are level, trend, and season. Winters' multiplicative exponential smoothing is not similar to any ARIMA model.
- Damped trend with multiplicative seasonal. This model is appropriate for a series in which there is a linear trend that is dying out and a seasonal effect that changes with the magnitude of the series. Its relevant smoothing parameters are level, trend, damping trend, and season. Damped trend and multiplicative seasonal exponential smoothing is not similar to any ARIMA model.
- Multiplicative trend with multiplicative seasonal. This model is appropriate for a series in which there are a trend and a seasonal effect that both change with the magnitude of the series. Its relevant smoothing parameters are level, trend, and season. Multiplicative trend and multiplicative seasonal exponential smoothing is not similar to any ARIMA model.

Target Transformation. You can specify a transformation to be performed on each dependent variable before it is modeled.

For more information, see [Series transformations](#).

- None. No transformation is performed.
- Square root. Square root transformation is performed.
- Natural log. Natural log transformation is performed.

ARIMA

Specify the structure of a custom ARIMA model.

ARIMA Orders. Enter values for the various ARIMA components of your model into the corresponding cells of the grid. All values must be non-negative integers. For autoregressive and moving average components, the value represents the maximum order. All positive lower orders are included in the model. For example, if you specify 2, the model includes orders 2 and 1. Cells in the Seasonal column are only enabled if a periodicity is defined for the active dataset.

- Autoregressive (p). The number of autoregressive orders in the model. Autoregressive orders specify which previous values from the series are used to predict current values. For example, an autoregressive order of 2 specifies that the value of the series two time periods in the past is used to predict the current value.
- Difference (d). Specifies the order of differencing applied to the series before estimating models. Differencing is necessary when trends are present (series with trends are typically nonstationary and ARIMA modeling assumes stationarity) and is used to remove their effect. The order of differencing corresponds to the degree of series trend; first-order differencing accounts for linear trends, second-order differencing accounts for quadratic trends, and so on.
- Moving Average (q). The number of moving average orders in the model. Moving average orders specify how deviations from the series mean for previous values are used to predict current values. For example, moving-average orders of 1 and 2 specify that deviations from the mean value of the series from each of the last two time periods be considered when predicting current values of the series.

Seasonal. Seasonal autoregressive, moving average, and differencing components play the same roles as their nonseasonal counterparts. For seasonal orders, however, current series values are affected by previous series values that are separated by one or more seasonal periods. For example, for monthly data (seasonal period of 12), a seasonal order of 1 means that the current series value is affected by the series value 12 periods before the current one. A seasonal order of 1, for monthly data, is then the same as specifying a nonseasonal order of 12.

Detect outliers automatically. Select this option to perform automatic detection of outliers, and select one or more of the outlier types available.

Type of Outliers to Detect. Select the outlier type(s) you want to detect. The supported types are:

- Additive (default)
- Level shift (default)
- Innovational
- Transient
- Seasonal additive
- Local trend
- Additive patch

Transfer Function Orders and Transformations. To specify transformations and to define transfer functions for any or all of the input fields in your ARIMA model, click Set; a separate dialog box is displayed in which you enter the transfer and transformation details.

Include constant in model. Inclusion of a constant is standard unless you are sure that the overall mean series value is 0. Excluding the constant is recommended when differencing is applied.

Further details

- For more information on types of outliers, see [Outliers](#).
- For more information on transfer and transformation functions, see [Transfer and transformation functions](#).
- [Transfer and transformation functions](#)

¹ Gardner, E. S. 1985. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4, 1-28.

Transfer and transformation functions

Use the Transfer Function Orders and Transformations dialog box to specify transformations and to define transfer functions for any or all of the input fields in your ARIMA model.

Target Transformations. In this pane you can specify a transformation to be performed on each target variable before it is modeled.

- None. No transformation is performed.
- Square root. Square root transformation is performed.
- Natural log. Natural log transformation is performed.

For more information, see [Series transformations](#).

Candidate Inputs Transfer functions and Transformation. You use transfer functions to specify the manner in which past values of the input fields are used to forecast future values of the target series. The list on the left hand side of the pane shows all input fields. The remaining information in this pane is specific to the input field you select.

Transfer Function Orders. Enter values for the various components of the transfer function into the corresponding cells of the Structure grid. All values must be non-negative integers. For numerator and denominator components, the value represents the maximum order. All positive lower orders are included in the model. In addition, order 0 is always included for numerator components. For example, if you specify 2 for numerator, the model includes orders 2, 1, and 0. If you specify 3 for denominator, the model includes orders 3, 2, and 1. Cells in the Seasonal column are only enabled if a periodicity is defined for the active dataset.

Numerator. The numerator order of the transfer function specifies which previous values from the selected independent (predictor) series are used to predict current values of the dependent series. For example, a numerator order of 1 specifies that the value of an independent series one time period in the past, in addition to the current value of the independent series, is used to predict the current value of each dependent series.

Denominator. The denominator order of the transfer function specifies how deviations from the series mean, for previous values of the selected independent (predictor) series, are used to predict current values of the dependent series. For example, a denominator order of 1 specifies that deviations from the mean value of an independent series one time period in the past is considered when predicting the current value of each dependent series.

Difference. Specifies the order of differencing applied to the selected independent (predictor) series before estimating models. Differencing is necessary when trends are present and is used to remove their effect.

Seasonal. Seasonal numerator, denominator, and differencing components play the same roles as their nonseasonal counterparts. For seasonal orders, however, current series values are affected by previous series values that are separated by one or more seasonal periods. For example, for monthly data (seasonal period of 12), a seasonal order of 1 means that the current series value is affected by the series value 12 periods prior to the current one. A seasonal order of 1, for monthly data, is then the same as specifying a nonseasonal order of 12.

Delay. Setting a delay causes the input field's influence to be delayed by the number of intervals specified. For example, if the delay is set to 5, the value of the input field at time t doesn't affect forecasts until five periods have elapsed ($t + 5$).

Transformation. Specification of a transfer function for a set of independent variables also includes an optional transformation to be performed on those variables.

- None. No transformation is performed.
- Square root. Square root transformation is performed.
- Natural log. Natural log transformation is performed.

Streaming Time Series node - model options

Confidence limit width (%). Confidence intervals are computed for the model predictions and residual autocorrelations. You can specify any positive value less than 100. By default, a 95% confidence interval is used.

The option to Extend records into the future sets the number of time intervals to forecast beyond the end of the estimation period. The time interval in this case is the time interval of the analysis, which you specify on the Data Specifications tab. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period. There is no maximum limit for this setting.

Future Values to Use in Forecasting

- Compute future values of inputs If you select this option, the forecast values for predictors, noise predictions, variance estimation, and future time values are calculated automatically. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period.
 - Select fields whose values you wish to add to the data. For each record that you want to forecast (excluding holdouts), if you are using predictor fields (with the role set to **Input**), you can specify estimated values for the forecast period for each predictor. You can either specify values manually, or choose from a list.
 - Field. Click the field selector button and choose any fields that may be used as predictors. Note that fields selected here may or may not be used in modeling; to actually use a field as a predictor, it must be selected in a downstream modeling node. This dialog box simply gives you a convenient place to specify future values so they can be shared by multiple downstream modeling nodes without specifying them separately in each node. Also note that the list of available fields may be constrained by selections on the Build Options tab.
- Note that if future values are specified for a field that is no longer available in the stream (because it has been dropped or because of updated selections made on the Build Options tab), the field is shown in red.
- Values. For each field, you can choose from a list of functions, or click Specify to either enter values manually or choose from a list of predefined values. If the predictor fields relate to items that are under your control, or which are otherwise knowable in advance, you should enter values manually. For example, if you are forecasting next month's revenues for a hotel based on the number of room reservations, you could specify the number of reservations you actually have for that period. Conversely, if a predictor field relates to something outside your control, such as a stock price, you could use a function such as the most recent value or the mean of recent points.
- The available functions depend on the measurement level of the field.

Table 1. Functions available for measurement levels

Measurement level	Functions
Continuous or Nominal field	Blank Mean of recent points Most recent value Specify
Flag field	Blank Most recent value True False Specify

Mean of recent points calculates the future value from the mean of the last three data points.

Most recent value sets the future value to that of the most recent data point.

True/False sets the future value of a flag field to True or False as specified.

Specify opens a dialog box for specifying future values manually, or choosing them from a predefined list.

Make Available for Scoring

You can set the default values here for the scoring options that appear on the dialog box for the model nugget.

- Calculate upper and lower confidence limits. If selected, this option creates new fields (with the default prefixes \$TSLCI- and \$TSUCI-) for the lower and upper confidence intervals, for each target field.
- Calculate noise residuals. If selected, this option creates a new field (with the default prefix \$TSResidual-) for the model residuals for each target field, together with a total of these values.

Model Settings

Maximum number of models to be displayed in output. Specify the maximum number of models you want to include in the output. Note that if the number of models built exceeds this threshold, the models are not shown in the output but they're still available for scoring. Default value is 10. Displaying a large number of models may result in poor performance or instability.

SMOTE node

The Synthetic Minority Over-sampling Technique (SMOTE) node provides an over-sampling algorithm to deal with imbalanced data sets. It provides an advanced method for balancing data. The SMOTE process node is implemented in Python and requires the imbalanced-learn® Python library. For details about the imbalanced-learn library, see <https://imbalanced-learn.org/stable/>¹.

The Python tab on the Nodes Palette contains the SMOTE node and other Python nodes.

¹Lemaître, Nogueira, Aridas. "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning." *Journal of Machine Learning Research*, vol. 18, no. 17, 2017, pp. 1-5. (<http://jmlr.org/papers/v18/16-365.html>)

- [SMOTE node Settings](#)
- [SMOTE node Settings](#)

SMOTE node Settings

Define the following settings on the SMOTE node's Settings tab.

Target Setting

Target Field. Select the target field. All flag, nominal, ordinal, and discrete measurement types are supported. If the Use partitioned data option is selected in the Partition section, only training data will be over-sampled.

Over Sample Ratio

Select Auto to automatically select an over-sample ratio, or select Set Ratio (minority over majority) to set a custom ratio value. The ratio is the number of samples in the minority class over the number of samples in the majority class. The value must be greater than 0 and less than or equal to 1.

Random Seed

Set random seed. Select this option and click Generate to generate the seed used by the random number generator.

Methods

Algorithm Kind. Select the type of SMOTE algorithm you wish to use.

Samples Rules

K Neighbours. Specify the number of the nearest neighbors to use for constructing synthetic samples

M Neighbours. Specify the number of nearest neighbors to use for determining if a minority sample is in danger. This will only be used if the Borderline1 or Borderline2 SMOTE algorithm type is selected.

Partition

Use partitioned data. Select this option if you only want training data to be over-sampled.

The SMOTE node requires the imbalanced-learn® Python library. The following table shows the relationship between the settings in the SPSS® Modeler SMOTE node dialog and the Python algorithm.

Table 1. Node properties mapped to Python library parameters

SPSS Modeler setting	Script name (property name)	Python API parameter name
Over sample ratio (number input control)	sample_ratio_value	ratio
Random seed	random_seed	random_state
K_Neighbours	k_neighbours	k
M_Neighbours	m_neighbours	m
Algorithm kind	algorithm_kind	kind

Extension Transform node

With the Extension Transform node, you can take data from an IBM® SPSS® Modeler stream and apply transformations to the data using R scripting or Python for Spark scripting. When the data has been modified, it is returned to the stream for further processing, model building and model scoring. The Extension Transform node makes it possible to transform data using algorithms that are written in R or Python for Spark, and enables the user to develop data transformation methods that are tailored to a particular problem.

To use this node with R, you must install IBM SPSS Modeler - Essentials for R. See the *IBM SPSS Modeler - Essentials for R: Installation Instructions* for installation instructions and compatibility information. You must also have a compatible version of R installed on your computer.

- [Extension Transform node - Syntax tab](#)
- [Extension Transform node - Console Output tab](#)

Extension Transform node - Syntax tab

Select your type of syntax – R or Python for Spark. See the following sections for more information. When your syntax is ready, you can click Run to execute the Extension Transform node.

R Syntax

R Syntax. You can enter, or paste, custom R scripting syntax for data analysis into this field.

Convert flag fields. Specifies how flag fields are treated. There are two options: Strings to factor, Integers and Reals to double, and Logical values (True, False). If you select Logical values (True, False) the original values of the flag fields are lost. For example, if a field has values Male and Female, these are changed to True and False.

Convert missing values to the R 'not available' value (NA). When selected, any missing values are converted to the R NA value. The value NA is used by R to identify missing values. Some R functions that you use might have an argument that can be used to control how the function behaves when the data contain NA. For example, the function might allow you to choose to automatically exclude records that contain NA. If this option is not selected, any missing values are passed to R unchanged, and might cause errors when your R script is executed.

Convert date/time fields to R classes with special control for time zones. When selected, variables with date or datetime formats are converted to R date/time objects. You must select one of the following options:

- **R POSIXct.** Variables with date or datetime formats are converted to R POSIXct objects.
- **R POSIXlt (list).** Variables with date or datetime formats are converted to R POSIXlt objects.

Note: The POSIX formats are advanced options. Use these options only if your R script specifies that datetime fields are treated in ways that require these formats. The POSIX formats do not apply to variables with time formats.

Python Syntax

Python Syntax. You can enter, or paste, custom Python scripting syntax for data analysis into this field. For more information about Python for Spark, see [Python for Spark](#) and [Scripting with Python for Spark](#).

Extension Transform node - Console Output tab

The Console Output tab contains any output that is received when the R script or Python for Spark script on the Syntax tab runs (for example, if using an R script, it shows output received from the R console when the R script in the R Syntax field on the Syntax tab is executed). This output might include R or Python error messages or warnings that are produced when the R or Python script is executed. The output can be used, primarily, to debug the script. The Console Output tab also contains the script from the R Syntax or Python Syntax field.

Every time the Extension Transform script is executed, the content of the Console Output tab is overwritten with the output received from the R console or Python for Spark. The output cannot be edited.

Space-Time-Boxes Node

Space-Time-Boxes (STB) are an extension of Geohashed spatial locations. More specifically, an STB is an alphanumeric string that represents a regularly shaped region of space and time.

For example, the STB dr5ru7|2013-01-01 00:00:00|2013-01-01 00:15:00 is made up of the following three parts:

- The geohash dr5ru7
- The start timestamp 2013-01-01 00:00:00
- The end timestamp 2013-01-01 00:15:00

As an example, you could use space and time information to improve confidence that two entities are the same because they are virtually in the same place at the same time. Alternatively, you could improve the accuracy of relationship identification by showing that two entities are related due to their proximity in space and time.

You can choose the Individual Records or Hangouts mode as appropriate for your requirements. Both modes require the same basic details, as follows:

Latitude field. Select the field that identifies the latitude (in WGS84 coordinate system).

Longitude field. Select the field that identifies the longitude (in WGS84 coordinate system).

Timestamp field. Select the field that identifies the time or date.

Individual Record Options

Use this mode to add an additional field to a record to identify its location at a given time.

Derive. Select one or more densities of space and time from which to derive the new field. See [Defining Space-Time-Box density](#) for more information.

Field name extension. Type the extension that you would like added to the new field name(s). You can choose to add this extension as either a Suffix or Prefix.

Hangout Options

A hangout can be thought of as a location and/or time in which an entity is continually or repeatedly found. For example, this could be used to identify a vehicle that makes regular transportation runs and identify any deviations from the norm.

The hangout detector monitors the movement of entities and flags conditions where an entity is observed to be "hanging out" in the area. The hangout detector automatically assigns each flagged hangout to one or more STBs, and uses in-memory entity and event tracking to detect hangouts with optimum efficiency.

STB Density. Select the density of space and time from which to derive the new field. For example, a value of STB_GH4_10MINS would correspond to a four-character geohash box of size approximately 20 km by 20 km and a 10-minute time window. See [Defining Space-Time-Box density](#) for more information.

Entity ID field. Select the entity to be used as the hangout identifier. This ID field identifies the event.

Minimum number of events. An event is a row in the data. Select the minimum number of occurrences of an event for the entity to be considered to be hanging out. A hangout must also qualify based on the following Dwell time is at least field.

Dwell time is at least. Specify the minimum duration over which the entity must dwell in the same location. This can help exclude, for example, a car waiting at a traffic light from being considered as hanging out. A hangout must also qualify based on the previous Minimum number of events field.

Following is more detail about what qualifies as a hangout:

Let e_1, \dots, e_n denote all time ordered events that are received from a given entity ID during a time duration (t_1, t_n). These events qualify as a hangout if:

- $n \geq \text{minimum number of events}$
- $t_n - t_1 \geq \text{minimum dwell time}$
- All events e_1, \dots, e_n occur in the same STB

Allow hangouts to span STB boundaries. If this option is selected the definition of a hangout is less strict and could include, for example, an entity that hangs out in more than one Space-Time-Box. For example, if your STBs are defined as whole hours, selecting this option would recognize an entity that hangs out for an hour as valid, even if the hour consisted of the 30 minutes before midnight and the 30 minutes after midnight. If this option is not selected, 100% of hangout time must be within a single Space-Time-Box.

Min proportion of events in qualifying timebox (%). Only available if Allow hangouts to span STB boundaries is selected. Use this to control the degree to which a hangout reported in one STB might in fact overlap another. Select the minimum proportion of events that must occur within a single STB to identify a hangout. If set to 25%, and the proportion of events is 26%, this qualifies as being a hangout.

For example, suppose you configure the hangout detector to require at least two events (minimum number of events = 2) and a contiguous hover time of at least 2 minutes in a 4-byte-geohash space box and a 10-minute time box (STB_NAME = STB_GH4_10MINS). When a hangout is detected, say the entity hovers in the same 4-byte-geohash space box while the three qualifying events occur within a 10-minute time span between 4:57pm and 5:07pm at 4:58pm, 5:01pm, and 5:03pm. The qualifying timebox percent value specifies the STBs that be credit for the hangout, as follows:

- 100%. Hangout is reported in the 5:00 - 5:10pm time-box and not in the 4:50 - 5:00pm time-box (events 5:01pm and 5:03pm meet all conditions that are required for a qualifying hangout and 100% of these events occurred in the 5:00 - 5:10 time box).
- 50%. Hangouts in both the time-boxes are reported (events 5:01pm and 5:03pm meet all conditions that are required for a qualifying hangout and at least 50% of these events occurred in the 4:50 - 5:00 time box and at least 50% of these events occurred in the 5:00 - 5:10 time box).
- 0%. Hangouts in both the time-boxes are reported.

When 0% is specified, hangout reports include the STBs representing every time box that is touched by the qualifying duration. The qualifying duration needs to be less than or equal to the corresponding duration of the time box in the STB. In other words, there should never be a configuration where a 10-minute STB is configured in tandem with a 20-minute qualifying duration.

A hangout is reported as soon as the qualifying conditions are met, and is not reported more than once per STB. Suppose that three events qualify for a hangout, and 10 total events occur within a qualifying duration all within the same STB. In that case, the hangout is reported when the third qualifying event occurs. None of the additional seven events trigger a hangout report.

Note:

- The hangout detector's in-memory event data is not shared across processes. Therefore, a particular entity has affinity with a particular hangout detector node. That is, incoming motion data for an entity must always be consistently passed to the hangout detector node tracking that entity, which is ordinarily the same node throughout the run.
- The hangout detector's in-memory event data is volatile. Whenever the hangout detector is exited and restarted, any work-in-progress hangouts are lost. This means stopping and restarting the process might cause the system to miss reporting real hangouts. A potential remedy involves replaying some of the historical motion data (for example, going back 48 hours and replaying the motion records that are applicable to any node that was restarted).
- The hangout detector must be fed data in time-sequential order.
- [Defining Space-Time-Box density](#)

Related information

- [Defining Space-Time-Box density](#)
-

Defining Space-Time-Box density

Choose the size (density) of your Space-Time-Boxes (STB) by specifying the physical area and elapsed time to be included in each.

Geo density. Select the size of the area to be included in each STB.

Time interval. Select the number of hours to be included in each STB.

Field name. Prefixed with STB, this is automatically completed based on your selections in the preceding two fields.

Streaming TCM Node

The Streaming TCM node can be used to build and score temporal causal models in one step.

For more information about temporal causal modeling, see [Temporal Causal Models](#).

- [Streaming TCM Node - Time Series options](#)
 - [Streaming TCM Node - Observations options](#)
 - [Streaming TCM Node - Time Interval options](#)
 - [Streaming TCM Node - Aggregation and Distribution options](#)
 - [Streaming TCM Node - Missing Value options](#)
 - [Streaming TCM Node - General Data options](#)
 - [Streaming TCM Node - General Build options](#)
 - [Streaming TCM Node - Estimation Period options](#)
 - [Streaming TCM Node - Model options](#)
-

Streaming TCM Node - Time Series options

On the Fields tab, use the Time Series settings to specify the series to include in the model system.

Select the option for the data structure that applies to your data. For multidimensional data, click Select Dimensions to specify the dimension fields. The specified order of the dimension fields defines the order in which they appear in all subsequent dialogs and output. Use the up and down arrow buttons on the Select Dimensions subdialog to reorder the dimension fields.

For column-based data, the term *series* has the same meaning as the term *field*. For multidimensional data, fields that contain time series are referred to as *metric* fields. A time series, for multidimensional data, is defined by a metric field and a value for each of the dimension fields. The following considerations apply to both column-based and multidimensional data.

- Series that are specified as candidate inputs or as both target and input are considered for inclusion in the model for each target. The model for each target always includes lagged values of the target itself.
- Series that are specified as forced inputs are always included in the model for each target.
- At least one series must be specified as either a target or as both target and input.
- When Use predefined roles is selected, fields that have a role of Input are set as candidate inputs. No predefined role maps to a forced input. For more information, see the topic [Roles](#).

Multidimensional data

For multidimensional data, you specify metric fields and associated roles in a grid, where each row in the grid specifies a single metric and role. By default, the model system includes series for all combinations of the dimension fields for each row in the grid. For example, if there are dimensions for *region* and *brand* then, by default, specifying the metric *sales* as a target means that there is a separate sales target series for each combination of *region* and *brand*.

For each row in the grid, you can customize the set of values for any of the dimension fields by clicking the ellipsis button for a dimension. This action opens the Select Dimension Values subdialog. You can also add, delete, or copy grid rows.

The Series Count column displays the number of sets of dimension values that are currently specified for the associated metric. The displayed value can be larger than the actual number of series (one series per set). This condition occurs when some of the specified combinations of dimension values do not correspond to series contained by the associated metric.

- [Streaming TCM Node - Select Dimension Values](#)
-

Streaming TCM Node - Select Dimension Values

For multidimensional data, you can customize the analysis by specifying which dimension values apply to a particular metric field with a particular role. For example, if *sales* is a metric field and *channel* is a dimension with values 'retail' and 'web,' you can specify that 'web' sales is an input and 'retail' sales is a target. You can also specify dimension subsets that apply to all metric fields used in the analysis. For example, if *region* is a dimension field that indicates geographical region, then you can limit the analysis to particular regions.

All values

Specifies that all values of the current dimension field are included. This option is the default.

Select values to include or exclude

Use this option to specify the set of values for the current dimension field. When **Include** is selected for the Mode, only values that are specified in the Selected values list are included. When **Exclude** is selected for the Mode, all values other than the values that are specified in the Selected values list are included.

You can filter the set of values from which to choose. Values that meet the filter condition appear in the Matched tab and values that do not meet the filter condition appear in the Unmatched tab of the Unselected values list. The All tab lists all unselected values, regardless of any filter condition.

- You can use asterisks (*) to indicate wildcard characters when you specify a filter.
 - To clear the current filter, specify an empty value for the search term on the Filter Displayed Values dialog.
-

Streaming TCM Node - Observations options

On the Fields tab, use the Observations settings to specify the fields that define the observations.

Observations that are defined by date/times

You can specify that the observations are defined by a date, time, or timestamp field. In addition to the field that defines the observations, select the appropriate time interval that describes the observations. Depending on the specified time interval, you can also specify other settings, such as the interval between observations (increment) or the number of days per week. The following considerations apply to the time interval:

- Use the value **Irregular** when the observations are irregularly spaced in time, such as the time at which a sales order is processed. When **Irregular** is selected, you must specify the time interval that is used for the analysis, from the Time Interval settings on the Data Specifications tab.
- When the observations represent a date and time and the time interval is hours, minutes, or seconds, then use Hours per day, Minutes per day, or Seconds per day. When the observations represent a time (duration) without reference to a date and the time interval is hours, minutes, or seconds, then use Hours (non-periodic), Minutes (non-periodic), or Seconds (non-periodic).
- Based on the selected time interval, the procedure can detect missing observations. Detecting missing observations is necessary since the procedure assumes that all observations are equally spaced in time and that there are no missing observations. For example, if the time interval is Days and the date 2014-10-27 is followed by 2014-10-29, then there is a missing observation for 2014-10-28. Values are imputed for any missing observations. Settings for handling missing values can be specified from the Data Specifications tab.
- The specified time interval allows the procedure to detect multiple observations in the same time interval that need to be aggregated together and to align observations on an interval boundary, such as the first of the month, to ensure that the observations are equally spaced. For example, if the time interval is Months, then multiple dates in the same month are aggregated together. This type of aggregation is referred to as *grouping*. By default, observations are summed when grouped. You can specify a different method for grouping, such as the mean of the observations, from the Aggregation and Distribution settings on the Data Specifications tab.
- For some time intervals, the additional settings can define breaks in the normal equally spaced intervals. For example, if the time interval is Days, but only weekdays are valid, you can specify that there are five days in a week, and the week begins on Monday.

Observations that are defined by periods or cyclic periods

Observations can be defined by one or more integer fields that represent periods or repeating cycles of periods, up to an arbitrary number of cycle levels. With this structure, you can describe series of observations that don't fit one of the standard time intervals. For example, a fiscal year with only 10 months can be described with a cycle field that represents years and a period field that represents months, where the length of one cycle is 10.

Fields that specify cyclic periods define a hierarchy of periodic levels, where the lowest level is defined by the Period field. The next highest level is specified by a cycle field whose level is 1, followed by a cycle field whose level is 2, and so on. Field values for each level, except the highest, must be periodic with respect to the next highest level. Values for the highest level cannot be periodic. For example, in the case of the 10-month fiscal year, months are periodic within years and years are not periodic.

- The length of a cycle at a particular level is the periodicity of the next lowest level. For the fiscal year example, there is only one cycle level and the cycle length is 10 since the next lowest level represents months and there are 10 months in the specified fiscal

- year.
- Specify the starting value for any periodic field that does not start from 1. This setting is necessary for detecting missing values. For example, if a periodic field starts from 2 but the starting value is specified as 1, then the procedure assumes that there is a missing value for the first period in each cycle of that field.
-

Streaming TCM Node - Time Interval options

The time interval that is used for the analysis can differ from the time interval of the observations. For example, if the time interval of the observations is Days, you might choose Months for the time interval for analysis. The data are then aggregated from daily to monthly data before the model is built. You can also choose to distribute the data from a longer to a shorter time interval. For example, if the observations are quarterly then you can distribute the data from quarterly to monthly data.

The available choices for the time interval at which the analysis is done depend on how the observations are defined and the time interval of those observations. In particular, when the observations are defined by cyclic periods, then only aggregation is supported. In that case, the time interval of the analysis must be greater than or equal to the time interval of the observations.

The time interval for the analysis is specified from the Time Interval settings on the Data Specifications tab. The method by which the data are aggregated or distributed is specified from the Aggregation and Distribution settings on the Data Specifications tab.

Streaming TCM Node - Aggregation and Distribution options

Aggregation functions

When the time interval that is used for the analysis is longer than the time interval of the observations, the input data are aggregated. For example, aggregation is done when the time interval of the observations is Days and the time interval for analysis is Months. The following aggregation functions are available: mean, sum, mode, min, or max.

Distribution functions

When the time interval that is used for the analysis is shorter than the time interval of the observations, the input data are distributed. For example, distribution is done when the time interval of the observations is Quarters and the time interval for analysis is Months. The following distribution functions are available: mean or sum.

Grouping functions

Grouping is applied when observations are defined by date/times and multiple observations occur in the same time interval. For example, if the time interval of the observations is Months, then multiple dates in the same month are grouped and associated with the month in which they occur. The following grouping functions are available: mean, sum, mode, min, or max. Grouping is always done when the observations are defined by date/times and the time interval of the observations is specified as Irregular.

Note: Although grouping is a form of aggregation, it is done before any handling of missing values whereas formal aggregation is done after any missing values are handled. When the time interval of the observations is specified as Irregular, aggregation is done only with the grouping function.

Aggregate cross-day observations to previous day

Specifies whether observations with times that cross a day boundary are aggregated to the values for the previous day. For example, for hourly observations with an eight-hour day that starts at 20:00, this setting specifies whether observations between 00:00 and 04:00 are included in the aggregated results for the previous day. This setting applies only if the time interval of the observations is Hours per day, Minutes per day or Seconds per day and the time interval for analysis is Days.

Custom settings for specified fields

You can specify aggregation, distribution, and grouping functions on a field by field basis. These settings override the default settings for the aggregation, distribution, and grouping functions.

Streaming TCM Node - Missing Value options

Missing values in the input data are replaced with an imputed value. The following replacement methods are available:

Linear interpolation

Replaces missing values by using a linear interpolation. The last valid value before the missing value and the first valid value after the missing value are used for the interpolation. If the first or last observation in the series has a missing value, then the two nearest non-missing values at the beginning or end of the series are used.

Series mean

Replaces missing values with the mean for the entire series.

Mean of nearby points

Replaces missing values with the mean of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the mean.

Median of nearby points

Replaces missing values with the median of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the median.

Linear trend

This option uses all non-missing observations in the series to fit a simple linear regression model, which is then used to impute the missing values.

Other settings:

Maximum percentage of missing values (%)

Specifies the maximum percentage of missing values that are allowed for any series. Series with more missing values than the specified maximum are excluded from the analysis.

Streaming TCM Node - General Data options

Maximum number of distinct values per dimension field

This setting applies to multidimensional data and specifies the maximum number of distinct values that are allowed for any one dimension field. By default, this limit is set to 10000 but it can be increased to an arbitrarily large number.

Streaming TCM Node - General Build options

Confidence interval width (%)

This setting controls the confidence intervals for both forecasts and model parameters. You can specify any positive value less than 100. By default, a 95% confidence interval is used.

Maximum number of inputs for each target

This setting specifies the maximum number of inputs that are allowed in the model for each target. You can specify an integer in the range 1 - 20. The model for each target always includes lagged values of itself, so setting this value to 1 specifies that the only input is the target itself.

Model tolerance

This setting controls the iterative process that is used for determining the best set of inputs for each target. You can specify any value that is greater than zero. The default is 0.001. Model tolerance is a stop criterion for predictor selection. It can affect the number of predictors that are included in the final model. But if a target can predict itself very well, other predictors may not be included in the final model.

Some trial and error may be required (for example, if you have this value set to high, you can try setting it to a smaller value to see if other predictors can be included or not).

Outlier threshold (%)

An observation is flagged as an outlier if the probability, as calculated from the model, that it is an outlier exceeds this threshold. You can specify a value in the range 50 - 100.

Number of Lags for Each Input

This setting specifies the number of lag terms for each input in the model for each target. By default, the number of lag terms is automatically determined from the time interval that is used for the analysis. For example, if the time interval is months (with an increment of one month) then the number of lags is 12. Optionally, you can explicitly specify the number of lags. The specified value must be an integer in the range 1 - 20.

Continue estimation using existing models

If you already generated a temporal causal model, select this option to reuse the criteria settings that are specified for that model, rather than building a new model. In this way, you can save time by reestimating and producing a new forecast that is based on the same model settings as before but using more recent data.

Streaming TCM Node - Estimation Period options

By default, the estimation period starts at the time of the earliest observation and ends at the time of the latest observation across all series.

By start and end times

You can specify both the start and end of the estimation period or you can specify just the start or just the end. If you omit the start or the end of the estimation period, the default value is used.

- If the observations are defined by a date/time field, then enter values for start and end in the same format that is used for the date/time field.
- For observations that are defined by cyclic periods, specify a value for each of the cyclic periods fields. Each field is displayed in a separate column.

By latest or earliest time intervals

Defines the estimation period as a specified number of time intervals that start at the earliest time interval or end at the latest time interval in the data, with an optional offset. In this context, the time interval refers to the time interval of the analysis. For example,

assume that the observations are monthly but the time interval of the analysis is quarters. Specifying Latest and a value of 24 for the Number of time intervals means the latest 24 quarters.

Optionally, you can exclude a specified number of time intervals. For example, specifying the latest 24 time intervals and 1 for the number to exclude means that the estimation period consists of the 24 intervals that precede the last one.

Streaming TCM Node - Model options

Model Name

You can specify a custom name for the model or accept the automatically generated name, which is *TCM*.

Forecast

The option to Extend records into the future sets the number of time intervals to forecast beyond the end of the estimation period. The time interval in this case is the time interval of the analysis, which is specified on the Data Specifications tab. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period. There is no maximum limit for this setting.

CPLEX Optimization node

The CPLEX Optimization node provides the ability to use complex mathematical (CPLEX) based optimization via an Optimization Programming Language (OPL) model file. This functionality was available in the IBM® Analytical Decision Management product that's no longer supported, but you can also use the CPLEX node in SPSS® Modeler without requiring IBM Analytical Decision Management.

For more information about CPLEX optimization and OPL, see the [IBM ILOG CPLEX Optimization Studio documentation](#).

The CPLEX Optimization node supports multiple data sources, or multiple dimensional incoming data. You can connect several nodes to the CPLEX Optimization node, and each prior node can be utilized to provide data to OPL model computing – set as individual tuple sets with individual field mappings.

When outputting the data generated by the CPLEX Optimization node, the original data from the data sources can be output together as single indexes, or as multiple dimensional indexes of the result.

The CPLEX Optimization node's decision variable doesn't support complex arrays.

Note:

- When running a stream containing a CPLEX Optimization node on **IBM SPSS Modeler Server**, by default the embedded Community edition CPLEX library is used. It has a limitation of 1000 variables and 1000 constraints. If you install the full edition of IBM ILOG CPLEX and want to use the full edition's CPLEX engine instead, which doesn't have such limitations, complete the following step for your platform.
 - On Windows, edit options.cfg and add the OPL library path. For example:

```
cplex_opl_lib_path="<CPLEX_path>\opl\bin\<Platform_dir>"
```

Where **<CPLEX_path>** is the CPLEX installation directory such as C:\Program Files\IBM\ILOG\CPLEX_Studio127, and **<Platform_dir>** is the platform-specific directory such as x64_win64.

- On Linux, edit modelersrv.sh and add the OPL library path. For example:

```
CPLEX_OPL_LIB_PATH=<CPLEX_path>/opl/bin/<Platform_dir>
```

Where **<CPLEX_path>** is the CPLEX installation directory such as /root/Libs_127_FullEdition/Linux_x86_64, and **<Platform_dir>** is the platform-specific directory such as x86-64_linux.

Note:

- When running a stream containing a CPLEX Optimization node in **SPSS Modeler Solution Publisher**, by default the embedded Community edition CPLEX library is used. It has a limitation of 1000 variables and 1000 constraints. If you install the full edition of IBM ILOG CPLEX and want to use the full edition's CPLEX engine instead, which doesn't have such limitations, complete the following step for your platform.
 - On Windows, add the OPL library path as a command line argument for modelerrun.exe. For example:

```
-o cplex_opl_lib_path="<CPLEX_path>\opl\bin\<Platform_dir>"
```

Where **<CPLEX_path>** is the CPLEX installation directory such as C:\Program Files\IBM\ILOG\CPLEX_Studio127, and **<Platform_dir>** is the platform-specific directory such as x64_win64.

- On Linux, edit modelerrun and add the OPL library path. For example:

```
CPLEX_OPL_LIB_PATH=<CPLEX_path>/opl/bin/<Platform_dir>
```

Where <CPLEX_path> is the CPLEX installation directory such as /root/Libs_127_FullEdition/Linux_x86_64, and <Platform_dir> is the platform-specific directory such as x86-64_linux.

- CPLEX isn't supported on macOS. You can use the node (add it to your stream, edit its properties, etc), but you can't run it.
- [Setting options for the CPLEX Optimization node](#)

Setting options for the CPLEX Optimization node

The Options tab of the CPLEX Optimization node contains the following fields.

OPL Model File. Select an Optimization Programming Language (OPL) model file.

OPL Model. After selecting an OPL model, the contents are shown here.

Input Data

On the Input Data tab, the Data Source drop-down lists all data sources (prior nodes) connected to the current CPLEX Optimization node. Selecting a data source from the drop-down refreshes the Input Mappings section below. Click Apply All Fields to generate all field mappings for the selected data source automatically. The Input Mappings table will be populated automatically.

Enter the tuple set name in the OPL model that corresponds to the incoming data. Then, if needed, verify that all tuple fields are mapped to data input fields according to their order in the tuple definition.

After setting the input mapping for a data source, you can select a different data source from the drop-down and repeat the process. The previous data source mappings will be saved automatically. Click Apply or OK when finished.

Other Data

On the Other Data tab, use the OPL Data section if you need to specify any other data for the optimization.

Output

When the output is a decision variable, it must take the prior data sources (incoming data) as indexes, and the indexes must be predefined in the Input Mappings section on the Input Data tab. No other type of decision variables are currently supported. The decision variable can have a single index or multiple indexes. SPSS® Modeler will output the CPLEX results with all or part of the original incoming data together, which is consistent with other SPSS Modeler nodes. The referred corresponding indexes must be specified in the Output Tuple field described below.

On the Output tab, choose the output mode (Raw Output or Decision Variable) and specify other options as appropriate. The Raw Output option will output the objective function value directly, regardless of name.

Objective Function Value Variable Name in OPL. This field is enabled if you selected the Decision Variable output mode. Enter the name of the objective function value variable from the OPL model.

Objective Function Value Field Name for Output. Enter the field name to use in the output. The default is _OBJECTIVE.

Output Tuple. Enter the name of the predefined tuple from the incoming data. This acts as the indexes of the decision variable and is expected to be output with the Variable Outputs. The Output Tuple should be consistent with the decision variable definition in the OPL. If there are multiple indexes, the tuple names must be joined by a comma (,).

Variable Outputs. Add one or more variables to include in the output.

Field Operations Nodes

- [Field Operations Overview](#)
- [Automated Data Preparation](#)
- [Type Node](#)
- [Filtering or renaming fields](#)
- [Derive node](#)
- [Filler node](#)
- [Reclassify Node](#)
- [Anonymize Node](#)
- [Binning Node](#)
- [RFM Analysis Node](#)

- [Ensemble Node](#)
- [Partition Node](#)
- [Set to Flag Node](#)
- [Restructure Node](#)
- [Transpose Node](#)
- [History Node](#)
- [Field Reorder Node](#)
- [Time Intervals Node](#)
- [Reprojection Node](#)
- [Time Intervals Node \(deprecated\)](#)

Field Operations Overview

After an initial data exploration, you will probably have to select, clean, or construct data in preparation for analysis. The Field Operations palette contains many nodes useful for this transformation and preparation.

For example, using a Derive node, you might create an attribute that is not currently represented in the data. Or you might use a Binning node to recode field values automatically for targeted analysis. You will probably find yourself using a Type node frequently—it allows you to assign a measurement level, values, and a modeling role for each field in the dataset. Its operations are useful for handling missing values and downstream modeling.

The Field Operations palette contains the following nodes:

	The Automated Data Preparation (ADP) node can analyze your data and identify fixes, screen out fields that are problematic or not likely to be useful, derive new attributes when appropriate, and improve performance through intelligent screening and sampling techniques. You can use the node in fully automated fashion, allowing the node to choose and apply fixes, or you can preview the changes before they are made and accept, reject, or amend them as desired.
	The Type node specifies field metadata and properties. For example, you can specify a measurement level (continuous, nominal, ordinal, or flag) for each field, set options for handling missing values and system nulls, set the role of a field for modeling purposes, specify field and value labels, and specify values for a field.
	The Filter node filters (discards) fields, renames fields, and maps fields from one source node to another.
	The Derive node modifies data values or creates new fields from one or more existing fields. It creates fields of type formula, flag, nominal, state, count, and conditional.
	The Ensemble node combines two or more model nuggets to obtain more accurate predictions than can be gained from any one model.
	The Filler node replaces field values and changes storage. You can choose to replace values based on a CLEM condition, such as @BLANK(@FIELD). Alternatively, you can choose to replace all blanks or null values with a specific value. A Filler node is often used together with a Type node to replace missing values.
	The Anonymize node transforms the way field names and values are represented downstream, thus disguising the original data. This can be useful if you want to allow other users to build models using sensitive data, such as customer names or other details.
	The Reclassify node transforms one set of categorical values to another. Reclassification is useful for collapsing categories or regrouping data for analysis.
	The Binning node automatically creates new nominal (set) fields based on the values of one or more existing continuous (numeric range) fields. For example, you can transform a continuous income field into a new categorical field containing groups of income as deviations from the mean. Once you have created bins for the new field, you can generate a Derive node based on the cut points.
	The Recency, Frequency, Monetary (RFM) Analysis node enables you to determine quantitatively which customers are likely to be the best ones by examining how recently they last purchased from you (recency), how often they purchased (frequency), and how much they spent over all transactions (monetary).
	The Partition node generates a partition field, which splits the data into separate subsets for the training, testing, and validation stages of model building.
	The Set to Flag node derives multiple flag fields based on the categorical values defined for one or more nominal fields.
	The Restructure node converts a nominal or flag field into a group of fields that can be populated with the values of yet another field. For example, given a field named <i>payment type</i> , with values of <i>credit</i> , <i>cash</i> , and <i>debit</i> , three new fields would be created (<i>credit</i> , <i>cash</i> , <i>debit</i>), each of which might contain the value of the actual payment made.
	The Transpose node swaps the data in rows and columns so that records become fields and fields become records.

	
	Use the Time Intervals node to specify intervals and derive a new time field for estimating or forecasting. A full range of time intervals is supported, from seconds to years.
	The History node creates new fields containing data from fields in previous records. History nodes are most often used for sequential data, such as time series data. Before using a History node, you may want to sort the data using a Sort node.
	The Field Reorder node defines the natural order used to display fields downstream. This order affects the display of fields in a variety of places, such as tables, lists, and the Field Chooser. This operation is useful when working with wide datasets to make fields of interest more visible.
	Within SPSS® Modeler, items such as the Expression Builder spatial functions, the Spatio-Temporal Prediction (STP) Node, and the Map Visualization Node use the projected coordinate system. Use the Reproject node to change the coordinate system of any data that you import that uses a geographic coordinate system.

Several of these nodes can be generated directly from the audit report created by a Data Audit node. See the topic [Generating Other Nodes for Data Preparation](#) for more information.

Related information

- [Automated Data Preparation](#)
- [Type Node](#)
- [Filtering or renaming fields](#)
- [Derive node](#)
- [Ensemble Node](#)
- [Filler node](#)
- [Anonymize Node](#)
- [Reclassify Node](#)
- [Binning Node](#)
- [RFM Analysis Node](#)
- [Partition Node](#)
- [Set to Flag Node](#)
- [Restructure Node](#)
- [Transpose Node](#)
- [Time Intervals Node](#)
- [History Node](#)
- [Field Reorder Node](#)
- [Reprojection Node](#)

Automated Data Preparation

Preparing data for analysis is one of the most important steps in any project—and traditionally, one of the most time consuming. Automated Data Preparation (ADP) handles the task for you, analyzing your data and identifying fixes, screening out fields that are problematic or not likely to be useful, deriving new attributes when appropriate, and improving performance through intelligent screening techniques. You can use the algorithm in fully **automatic** fashion, allowing it to choose and apply fixes, or you can use it in **interactive** fashion, previewing the changes before they are made and accept or reject them as you want.

Using ADP enables you to make your data ready for model building quickly and easily, without needing prior knowledge of the statistical concepts involved. Models will tend to build and score more quickly; in addition, using ADP improves the robustness of automated modeling processes, such as model refresh and champion / challenger.

Note: When ADP prepares a field for analysis, it creates a new field containing the adjustments or transformations, rather than replacing the existing values and properties of the old field. The old field is not used in further analysis; its role is set to None.

Example. An insurance company with limited resources to investigate homeowner's insurance claims wants to build a model for flagging suspicious, potentially fraudulent claims. Before building the model, they will ready the data for modeling using automated data preparation. Since they want to be able to review the proposed transformations before the transformations are applied, they will use automated data preparation in interactive mode.

An automotive industry group keeps track of the sales for a variety of personal motor vehicles. In an effort to be able to identify over- and underperforming models, they want to establish a relationship between vehicle sales and vehicle characteristics. They will use automated data preparation to prepare the data for analysis, and build models using the data "before" and "after" preparation to see how the results differ.

What is your objective? Automated data preparation recommends data preparation steps that will affect the speed with which other algorithms can build models and improve the predictive power of those models. This can include transforming, constructing and selecting features. The target can also be transformed. You can specify the model-building priorities that the data preparation process should concentrate on.

- Balance speed and accuracy. This option prepares the data to give equal priority to both the speed with which data are processed by model-building algorithms and the accuracy of the predictions.
- Optimize for speed. This option prepares the data to give priority to the speed with which data are processed by model-building algorithms. When you are working with very large datasets, or are looking for a quick answer, select this option.
- Optimize for accuracy. This option prepares the data to give priority to the accuracy of predictions produced by model-building algorithms.
- Custom analysis. When you want to manually change the algorithm on the Settings tab, select this option. Note that this setting is automatically selected if you subsequently make changes to options on the Settings tab that are incompatible with one of the other objectives.

Training the node

The ADP node is implemented as a process node and works in a similar way to the Type node; **training** the ADP node corresponds to instantiating the Type node. Once analysis has been performed, the specified transformations are applied to the data without further analysis as long as the upstream data model does not change. Like the Type and Filter nodes, if the ADP node is disconnected it remembers the data model and transformations so that if it is reconnected it does not need to be retrained; this enables you to train it on a subset of typical data and then copy or deploy it for use on live data as often as required.

Using the toolbar

The toolbar enables you to run and update the display of the data analysis, and generate nodes that you can use in conjunction with the original data.

- Generate From this menu you can generate either a Filter or Derive node. Note that this menu is only available when there is an analysis shown on the Analysis tab.

The Filter node removes transformed input fields. If you configure the ADP node to leave the original input fields in the dataset, this will restore the original set of inputs allowing you to interpret the score field in terms of the inputs. For example, this may be useful if you want to produce a graph of the score field against various inputs.

The Derive node can restore the original dataset and target units. You can only generate a Derive node when the ADP node contains an analysis which rescales a range target (that is, Box-Cox rescaling is selected on the Prepare Inputs & Target panel). You cannot generate a Derive node if the target is not a range, or if the Box-Cox rescaling is not selected. See the topic [Generating a Derive node](#) for more information.

- View Contains options that control what is shown on the Analysis tab. This includes the graph editing controls and the display selections for both the main panel and linked views.
- Preview Displays a sample of the transformations that will be applied to the input data.
- Analyze Data Initiates an analysis using the current settings and displays the results on the Analysis tab.
- Clear Analysis Deletes the existing analysis (only available when a current analysis exists).

Node status

The status of the ADP node on the IBM® SPSS® Modeler canvas is indicated by either an arrow or tick on the icon that shows whether or not analysis has taken place.

For more information about the calculations performed with the Automated Data Preparation node, see the *Automated Data Preparation algorithms* section of the *IBM SPSS Modeler Algorithms Guide*. The guide is available in PDF format from the \Documentation directory of the installation disk, as part of your product download, or on the Web.

- [Fields tab \(automated data preparation\)](#)
- [Settings tab \(automated data preparation\)](#)
- [Analysis tab \(automated data preparation\)](#)
- [Generating a Derive node](#)

Fields tab (automated data preparation)

Before you can build a model, you need to specify which fields you want to use as targets and as inputs. With a few exceptions, all modeling nodes will use field information from an upstream Type node. If you are using a Type node to select input and target fields, you don't need to change anything on this tab.

Use type node settings. This option tells the node to use field information from an upstream Type node. This is the default.

Use custom settings. This option tells the node to use field information specified here instead of that given in any upstream Type node(s). After selecting this option, specify the fields below as required.

Target. For models that require one or more target fields, select the target field or fields. This is similar to setting the field role to *Target* in a Type node.

Inputs. Select the input field or fields. This is similar to setting the field role to *Input* in a Type node.

Settings tab (automated data preparation)

The Settings tab comprises several different groups of settings that you can modify to fine-tune how the algorithm processes your data. If you make any changes to the default settings that are incompatible with the other objectives, the Objective tab is automatically updated to select the Customize analysis option.

- [Field settings](#)
- [Prepare dates & times \(automated data preparation\)](#)
- [Excluding fields \(automated data preparation\)](#)
- [Preparing inputs and targets](#)
- [Construction and feature selection](#)
- [Field names \(automated data preparation\)](#)

Field settings

Use frequency field. This option enables you to select a field as a frequency weight. Use this if the records in your training data represent more than one unit each; for example, if you are using aggregated data. The field values should be the number of units represented by each record.

Use weight field. This option enables you to select a field as a case weight. Case weights are used to account for differences in variance across levels of the output field.

How to handle fields that are excluded from modeling. Specify what happens to excluded fields; you can choose either to filter them out of the data or simply set their *Role* to None.

Note: This action will also be applied to the target if it is transformed. For example, if the new derived version of the target is used as the Target field, the original target is either filtered or set to None.

If the incoming fields do not match the existing analysis. Specify what happens if one or more required input fields are missing from the incoming dataset when you execute a trained ADP node.

- Stop execution and keep the existing analysis. This stops the execution process, preserves the current analysis information, and displays an error.
- Clear the existing analysis and analyze the new data. This clears the existing analysis, analyzes the incoming data, and applies the recommended transformations to that data.

Prepare dates & times (automated data preparation)

Many modeling algorithms are unable to directly handle date and time details; these settings enable you to derive new duration data that can be used as model inputs from dates and times in your existing data. The fields containing dates and times must be predefined with date or time storage types. The original date and time fields will not be recommended as model inputs following automated data preparation.

Prepare dates and times for modeling. Deselecting this option disables all other Prepare Dates & Times controls while maintaining the selections.

Compute elapsed time until reference date. This produces the number of years/months/days since a reference date for each variable containing dates.

- Reference Date. Specify the date from which the duration will be calculated with regard to the date information in the input data. Selecting Today's date means that the current system date is always used when ADP is executed. To use a specific date, select Fixed date and enter the required date. The current date is automatically entered in the Fixed date field when the node is first created.
- Units for Date Duration. Specify whether ADP should automatically decide on the date duration unit, or select from Fixed units of Years, Months, or Days.

Compute elapsed time until reference time. This produces the number of hours/minutes/seconds since a reference time for each variable containing times.

- Reference Time. Specify the time from which the duration will be calculated with regard to the time information in the input data. Selecting Current time means that the current system time is always used when ADP is executed. To use a specific time, select Fixed time and enter the required details. The current time is automatically entered in the Fixed time field when the node is first created.
- Units for Time Duration. Specify whether ADP should automatically decide on the time duration unit, or select from Fixed units of Hours, Minutes, or Seconds.

Extract Cyclical Time Elements. Use these settings to split a single date or time field into one or more fields. For example if you select all three date checkboxes, the input date field "1954-05-23" is split into three fields: 1954, 5, and 23, each using the suffix defined on the Field Names panel, and the original date field is ignored.

- Extract from dates. For any date inputs, specify if you want to extract years, months, days, or any combination.
- Extract from times. For any time inputs, specify if you want to extract hours, minutes, seconds, or any combination.

Excluding fields (automated data preparation)

Poor quality data can affect the accuracy of your predictions; therefore, you can specify the acceptable quality level for input features. All fields that are constant or have 100% missing values are automatically excluded.

Exclude low quality input fields. Deselecting this option disables all other Exclude Fields controls while maintaining the selections.

Exclude fields with too many missing values. Fields with more than the specified percentage of missing values are removed from further analysis. Specify a value greater than or equal to 0, which is equivalent to deselecting this option, and less than or equal to 100, though fields with all missing values are automatically excluded. The default is 50.

Exclude nominal fields with too many unique categories. Nominal fields with more than the specified number of categories are removed from further analysis. Specify a positive integer. The default is 100. This is useful for automatically removing fields containing record-unique information from modeling, like ID, address, or name.

Exclude categorical fields with too many values in a single category. Ordinal and nominal fields with a category that contains more than the specified percentage of the records are removed from further analysis. Specify a value greater than or equal to 0, equivalent to deselecting this option, and less than or equal to 100, though constant fields are automatically excluded. The default is 95.

Preparing inputs and targets

Because no data is ever in a perfect state for processing, you may want to adjust some of the settings before running an analysis. For example, this might include the removal of outliers, specifying how to handle missing values, or adjusting the type.

Note: If you change the values on this panel, the Objectives tab is automatically updated to select the Custom analysis option. Prepare the input and target fields for modeling. Toggles all fields on the panel either on or off.

Adjust Type and Improve Data Quality. For inputs and the target you can specify several data transformations separately; this is because you may not want to change the values of the target. For example, a prediction of income in dollars is more meaningful than a prediction measured in log(dollars). In addition, if the target has missing values there is no predictive gain to filling missing values, whereas filling missing values in inputs may enable some algorithms to process information that would otherwise be lost.

Additional settings for these transformations, such as the outlier cutoff value, are common to both the target and inputs.

You can select the following settings for either, or both, inputs and target:

- Adjust the type of numeric fields. Select this to determine if numeric fields with a measurement level of *Ordinal* can be converted to *Continuous*, or vice versa. You can specify the minimum and maximum threshold values to control the conversion.
- Reorder nominal fields. Select this to sort nominal (set) fields into order, from smallest to largest category.
- Replace outlier values in continuous fields. Specify whether to replace outliers; use this in conjunction with the Method for replacing outliers options below.
- Continuous fields: replace missing values with mean. Select this to replace missing values of continuous (range) features.
- Nominal fields: replace missing values with mode. Select this to replace missing values of nominal (set) features.
- Ordinal fields: replace missing values with median. Select this to replace missing values of ordinal (ordered set) features.

Maximum number of values for ordinal fields. Specify the threshold for redefining ordinal (ordered set) fields as continuous (range). The default is 10; therefore, if an ordinal field has more than 10 categories it is redefined as continuous (range).

Minimum number of values for continuous fields. Specify the threshold for redefining scale or continuous (range) fields as ordinal (ordered set). The default is 5; therefore, if a continuous field has fewer than 5 values it is redefined as ordinal (ordered set).

Outlier cutoff value. Specify the outlier cutoff criterion, measured in standard deviations; the default is 3.

Method for replacing outliers. Select whether outliers are to be replaced by either trimming (coerce) with the cutoff value, or to delete them and set them as missing values. Any outliers set to missing values follow the missing value handling settings selected above.

Put all continuous input fields on a common scale. To normalize continuous input fields, select this check box and choose the normalization method. The default is z-score transformation, where you can specify the Final mean, which has a default of 0, and the Final standard deviation, which has a default of 1. Alternatively, you can choose to use Min/max transformation and specify the minimum and maximum values, which have default values of 0 and 100 respectively.

This field is especially useful when you select Perform feature construction on the Construct & Select Features panel.

Rescale a continuous target with a Box-Cox transformation. To normalize a continuous (scale or range) target field, select this check box. The Box-Cox transformation has default values of 0 for the Final mean and 1 for the Final standard deviation.

Note: If you choose to normalize the target, the dimension of the target will be transformed. In this case you may need to generate a Derive node to apply an inverse transformation in order to turn the transformed units back into a recognizable format for further processing. See the topic [Generating a Derive node](#) for more information.

Construction and feature selection

To improve the predictive power of your data, you can transform the input fields, or construct new ones based on the existing fields.

Note: If you change the values on this panel, the Objectives tab is automatically updated to select the Custom analysis option. Transform, construct and select input fields to improve predictive power. Toggles all fields on the panel either on or off.

Merge sparse categories to maximize association with target. Select this to make a more parsimonious model by reducing the number of variables to be processed in association with the target. If required, change the probability value from the default of 0.05.

Note that if all categories are merged into one, the original and derived versions of the field are excluded because they have no value as a predictor.

When there is no target, merge sparse categories based on counts. If you are dealing with data that has no target, you can choose to merge sparse categories of either, or both, ordinal (ordered set) and nominal (set) features. Specify the minimum percentage of cases, or records, in the data that identifies the categories to be merged; the default is 10.

Categories are merged using the following rules:

- Merging is not performed on binary fields.
- If there are only two categories during merging, merging stops.
- If there is no original category, nor any category created during merging, with fewer than the specified minimum percent of cases, merging stops.

Bin continuous fields while preserving predictive power. Where you have data that includes a categorical target, you can bin continuous inputs with strong associations to improve processing performance. If required, change the probability value for the homogenous subsets from the default of 0.05.

If the binning operation results in a single bin for a particular field, the original and binned versions of the field are excluded because they have no value as a predictor.

Note: Binning in ADP differs from optimal binning used in other parts of IBM® SPSS® Modeler. Optimal binning uses entropy information to convert a continuous variable to a categorical variable; this needs to sort data and store it all in memory. ADP uses homogenous subsets to bin a continuous variable, this means that ADP binning does not need to sort data and does not store all data in memory. The use of the homogenous subset method to bin a continuous variable means that the number of categories after binning is always less than or equal to the number of categories of target.

Perform feature selection. Select this option to remove features with a low correlation coefficient. If required, change the probability value from the default of 0.05.

This option only applies to continuous input features where the target is continuous, and to categorical input features.

Perform feature construction. Select this option to derive new features from a combination of several existing features (which are then discarded from modeling).

This option only applies to continuous input features where the target is continuous, or where there is no target.

Field names (automated data preparation)

To easily identify new and transformed features, ADP creates and applies basic new names, prefixes, or suffixes. You can amend these names to be more relevant to your own needs and data. If you want to specify other labels, you will need to do this in a downstream Type node.

Transformed and Constructed Fields. Specify the name extensions to be applied to transformed target and input fields.

Note that in the ADP node, setting string fields to contain nothing may cause an error depending on how you chose to handle unused fields. If How to handle fields that are excluded from modeling is set to Filter out unused fields on the Field Settings panel of the Settings tab, the name extensions for inputs and the target can be set to nothing. The original fields are filtered out and the transformed fields saved over them; in this case the new transformed fields will have the same name as your original.

However if you chose Set the direction of unused fields to 'None', then empty, or null, name extensions for the target and inputs will cause an error because you will be trying to create duplicate field names.

In addition, specify the prefix name to be applied to any features that are constructed via the Select and Construct settings. The new name is created by attaching a numeric suffix to this prefix root name. The format of the number depends on how many new features are derived, for example:

- 1-9 constructed features will be named: feature1 to feature9.
- 10-99 constructed features will be named: feature01 to feature99.
- 100-999 constructed features will be named: feature001 to feature999, and so on.

This ensures that the constructed features will sort in a sensible order no matter how many there are.

Durations Computed from Dates and Times. Specify the name extensions to be applied to durations computed from both dates and times.

Cyclical Elements Extracted from Dates and Times. Specify the name extensions to be applied to cyclical elements extracted from both dates and times.

Analysis tab (automated data preparation)

1. When you are satisfied with the ADP settings, including any changes made on the Objective, Fields, and Settings tabs, click Analyze Data; the algorithm applies the settings to the data inputs and displays the results on the Analysis tab.

The Analysis tab contains both tabular and graphical output that summarizes the processing of your data and displays recommendations as to how the data may be modified or improved for scoring. You can then review and either accept or reject those recommendations.

The Analysis tab is made up of two panels, the main view on the left and the linked, or auxiliary, view on the right. There are three main views:

- Field Processing Summary (the default). See the topic [Field Processing Summary \(automated data preparation\)](#) for more information.
- Fields. See the topic [Fields \(automated data preparation\)](#) for more information.
- Action Summary. See the topic [Action Summary \(automated data preparation\)](#) for more information.

There are four linked/auxiliary views:

- Predictive Power (the default). See the topic [Predictive Power \(automated data preparation\)](#) for more information.
- Fields Table. See the topic [Fields Table \(automated data preparation\)](#) for more information.
- Field Details. See the topic [Field Details \(automated data preparation\)](#) for more information.
- Action Details. See the topic [Action Details \(automated data preparation\)](#) for more information.

Links between views

Within the main view, underlined text in the tables controls the display in the linked view. Clicking on the text allows you to get details on a particular field, set of fields, or processing step. The link that you last selected is shown in a darker color; this helps you identify the connection between the contents of the two view panels.

Resetting the views

To redisplay the original Analysis recommendations and abandon any changes you have made to the Analysis views, click Reset at the bottom of the main view panel.

- [Field Processing Summary \(automated data preparation\)](#)
- [Fields \(automated data preparation\)](#)
- [Action Summary \(automated data preparation\)](#)
- [Predictive Power \(automated data preparation\)](#)
- [Fields Table \(automated data preparation\)](#)
- [Field Details \(automated data preparation\)](#)
- [Action Details \(automated data preparation\)](#)

Field Processing Summary (automated data preparation)

The Field Processing Summary table gives a snapshot of the projected overall impact of processing, including changes to the state of the features and the number of features constructed.

Note that no model is actually built, so there isn't a measure or graph of the change in overall predictive power before and after data preparation; instead, you can display graphs of the predictive power of individual recommended predictors.

The table displays the following information:

- The number of target fields.
- The number of original (input) predictors.
- The predictors recommended for use in analysis and modeling. This includes the total number of fields recommended; the number of original, untransformed, fields recommended; the number of transformed fields recommended (excluding intermediate versions of any field, fields derived from date/time predictors, and constructed predictors); the number of fields recommended that are derived from date/time fields; and the number of constructed predictors recommended.
- The number of input predictors not recommended for use in any form, whether in their original form, as a derived field, or as input to a constructed predictor.

Where any of the Fields information is underlined, click to display more details in a linked view. Details of the Target, Input features, and Input features not used are shown in the Fields Table linked view. See the topic [Fields Table \(automated data preparation\)](#) for more information. Features recommended for use in analysis are displayed in the Predictive Power linked view. See the topic [Predictive Power \(automated data preparation\)](#) for more information.

Fields (automated data preparation)

The Fields main view displays the processed fields and whether ADP recommends using them in downstream models. You can override the recommendation for any field; for example, to exclude constructed features or include features that ADP recommends excluding. If a field has been transformed, you can decide whether to accept the suggested transformation or use the original version.

The Fields view consists of two tables, one for the target and one for predictors that were either processed or created.

Target table

The Target table is only shown if a target is defined in the data.

The table contains two columns:

- Name. This is the name or label of the target field; the original name is always used, even if the field has been transformed.
 - Measurement Level. This displays the icon representing the measurement level; hover the mouse over the icon to display a label (continuous, ordinal, nominal, and so on) that describes the data.
- If the target has been transformed the Measurement Level column reflects the final transformed version. *Note:* you cannot turn off transformations for the target.

Predictors table

The Predictors table is always shown. Each row of the table represents a field. By default the rows are sorted in descending order of predictive power.

For ordinary features, the original name is always used as the row name. Both original and derived versions of date/time fields appear in the table (in separate rows); the table also includes constructed predictors.

Note that transformed versions of fields shown in the table always represent the final versions.

By default only recommended fields are shown in the Predictors table. To display the remaining fields, select the Include nonrecommended fields in table box above the table; these fields are then displayed at the bottom of the table.

The table contains the following columns:

- Version to Use. This displays a drop-down list that controls whether a field will be used downstream and whether to use the suggested transformations. By default, the drop-down list reflects the recommendations.
- For ordinary predictors that have been transformed the drop-down list has three choices: Transformed, Original, and Do not use.

For untransformed ordinary predictors the choices are: Original and Do not use.

For derived date/time fields and constructed predictors the choices are: Transformed and Do not use.

For original date fields the drop-down list is disabled and set to Do not use.

Note: For predictors with both original and transformed versions, changing between Original and Transformed versions automatically updates the Measurement Level and Predictive Power settings for those features.

- Name. Each field's name is a link. Click a name to display more information about the field in the linked view. See the topic [Field Details \(automated data preparation\)](#) for more information.
- Measurement Level. This displays the icon representing the data type; hover the mouse over the icon to display a label (continuous, ordinal, nominal, and so on) that describes the data.

- Predictive Power. Predictive power is displayed only for fields that ADP recommends. This column is not displayed if there is no target defined. Predictive power ranges from 0 to 1, with larger values indicating "better" predictors. In general, predictive power is useful for comparing predictors within an ADP analysis, but predictive power values should not be compared across analyses.

Action Summary (automated data preparation)

For each action taken by automated data preparation, input predictors are transformed and/or filtered out; fields that survive one action are used in the next. The fields that survive through to the last step are then recommended for use in modeling, whilst inputs to transformed and constructed predictors are filtered out.

The Action Summary is a simple table that lists the processing actions taken by ADP. Where any Action is underlined, click to display more details in a linked view about the actions taken. See the topic [Action Details \(automated data preparation\)](#) for more information.

Note: Only the original and final transformed versions of each field are shown, not any intermediate versions that were used during analysis.

Predictive Power (automated data preparation)

Displayed by default when the analysis is first run, or when you select Predictors recommended for use in analysis in the Field Processing Summary main view, the chart displays the predictive power of recommended predictors. Fields are sorted by predictive power, with the field with the highest value appearing at the top.

For transformed versions of ordinary predictors, the field name reflects your choice of suffix in the Field Names panel of the Settings tab; for example: *_transformed*.

Measurement level icons are displayed after the individual field names.

The predictive power of each recommended predictor is computed from either a linear regression or naïve Bayes model, depending upon whether the target is continuous or categorical.

Fields Table (automated data preparation)

Displayed when you click Target, Predictors, or Predictors not used in the Field Processing Summary main view, the Fields Table view displays a simple table listing the relevant features.

The table contains two columns:

- Name. The predictor name.
For targets, the original name or label of the field is used, even if the target has been transformed.

For transformed versions of ordinary predictors, the name reflects your choice of suffix in the Field Names panel of the Settings tab; for example: *_transformed*.

For fields derived from dates and times, the name of the final transformed version is used; for example: *bdate_years*.

For constructed predictors, the name of the constructed predictor is used; for example: *Predictor1*.
- Measurement Level. This displays the icon representing the data type.
For the Target, the Measurement Level always reflects the transformed version (if the target has been transformed); for example, changed from ordinal (ordered set) to continuous (range, scale), or vice versa.

Field Details (automated data preparation)

Displayed when you click any Name in the Fields main view, the Field Details view contains distribution, missing values, and predictive power charts (if applicable) for the selected field. In addition, the processing history for the field and the name of the transformed field are also shown (if applicable).

For each chart set, two versions are shown side by side to compare the field with and without transformations applied; if a transformed version of the field does not exist, a chart is shown for the original version only. For derived date or time fields and constructed predictors, the charts are only shown for the new predictor.

Note: If a field is excluded due to having too many categories only the processing history is shown.

Distribution Chart

Continuous field distribution is shown as a histogram, with a normal curve overlaid, and a vertical reference line for the mean value; categorical fields are displayed as a bar chart.

Histograms are labeled to show standard deviation and skewness; however, skewness is not displayed if the number of values is 2 or fewer or the variance of the original field is less than 10-20.

Hover the mouse over the chart to display either the mean for histograms, or the count and percentage of the total number of records for categories in bar charts.

Missing Value Chart

Pie charts compare the percentage of missing values with and without transformations applied; the chart labels show the percentage.

If ADP carried out missing value handling, the post-transformation pie chart also includes the replacement value as a label -- that is, the value used in place of missing values.

Hover the mouse over the chart to display the missing value count and percentage of the total number of records.

Predictive Power Chart

For recommended fields, bar charts display the predictive power before and after transformation. If the target has been transformed, the calculated predictive power is in respect to the transformed target.

Note: Predictive power charts are not shown if no target is defined, or if the target is clicked in the main view panel.

Hover the mouse over the chart to display the predictive power value.

Processing History Table

The table shows how the transformed version of a field was derived. Actions taken by ADP are listed in the order in which they were carried out; however, for certain steps multiple actions may have been carried out for a particular field.

Note: This table is not shown for fields that have not been transformed.

The information in the table is broken down into two or three columns:

- Action. The name of the action. For example, Continuous Predictors. See the topic [Action Details \(automated data preparation\)](#) for more information.
- Details. The list of processing carried out. For example, Transform to standard units.
- Function. Only shown for constructed predictors, this displays the linear combination of input fields, for example, .06*age + 1.21*height.

Action Details (automated data preparation)

Displayed when you select any underlined Action in the Action Summary main view, the Action Details linked view displays both action-specific and common information for each processing step that was carried out; the action-specific details are displayed first.

For each action, the description is used as the title at the top of the linked view. The action-specific details are displayed below the title, and may include details of the number of derived predictors, fields recast, target transformations, categories merged or reordered, and predictors constructed or excluded.

As each action is processed, the number of predictors used in the processing may change, for example as predictors are excluded or merged.

Note: If an action was turned off, or no target was specified, an error message is displayed in place of the action details when the action is clicked in the Action Summary main view.

There are nine possible actions; however, not all are necessarily active for every analysis.

Text Fields table

The table displays the number of:

- Trailing blank values trimmed.
- Predictors excluded from analysis.

Date and Time Predictors table

The table displays the number of:

- Durations derived from date and time predictors.
- Date and time elements.
- Derived date and time predictors, in total.

The reference date or time is displayed as a footnote if any date durations were calculated.

Predictor Screening table

The table displays the number of the following predictors excluded from processing:

- Constants.
- Predictors with too many missing values.
- Predictors with too many cases in a single category.
- Nominal fields (sets) with too many categories.
- Predictors screened out, in total.

Check Measurement Level table

The table displays the numbers of fields recast, broken down into the following:

- Ordinal fields (ordered sets) recast as continuous fields.
- Continuous fields recast as ordinal fields.
- Total number recast.

If no input fields (target or predictors) were continuous or ordinal, this is shown as a footnote.

Outliers table

The table displays counts of how any outliers were handled.

- Either the number of continuous fields for which outliers were found and trimmed, or the number of continuous fields for which outliers were found and set to missing, depending on your settings in the Prepare Inputs & Target panel on the Settings tab.
- The number of continuous fields excluded because they were constant, after outlier handling.

One footnote shows the outlier cutoff value; while another footnote is shown if no input fields (target or predictors) were continuous.

Missing Values table

The table displays the numbers of fields that had missing values replaced, broken down into:

- Target. This row is not shown if no target is specified.
- Predictors. This is further broken down into the number of nominal (set), ordinal (ordered set), and continuous.
- The total number of missing values replaced.

Target table

The table displays whether the target was transformed, shown as:

- Box-Cox transformation to normality. This is further broken down into columns that show the specified criteria (mean and standard deviation) and Lambda.
- Target categories reordered to improve stability.

Categorical Predictors table

The table displays the number of categorical predictors:

- Whose categories were reordered from lowest to highest to improve stability.
- Whose categories were merged to maximize association with the target.
- Whose categories were merged to handle sparse categories.
- Excluded due to low association with the target.
- Excluded because they were constant after merging.

A footnote is shown if there were no categorical predictors.

Continuous Predictors table

There are two tables. The first displays one of the following number of transformations:

- Predictor values transformed to standard units. In addition, this shows the number of predictors transformed, the specified mean, and the standard deviation.
- Predictor values mapped to a common range. In addition, this shows the number of predictors transformed using a min-max transformation, as well as the specified minimum and maximum values.
- Predictor values binned and the number of predictors binned.

The second table displays the predictor space construction details, shown as the number of predictors:

- Constructed.
- Excluded due to a low association with the target.
- Excluded because they were constant after binning.
- Excluded because they were constant after construction.

A footnote is shown if no continuous predictors were input.

Generating a Derive node

When you generate a Derive node, it applies the inverse target transformation to the score field. By default the node enters the name of the score field that would be produced by an automodeler (such as Auto Classifier or Auto Numeric) or the Ensemble node. If a scale (range) target has been transformed the score field is shown in transformed units; for example, `log($)` instead of `$`. In order to interpret and use the results, you must convert the predicted value back to the original scale.

Note: You can only generate a Derive node when the ADP node contains an analysis which rescales a range target (that is, Box-Cox rescaling is selected on the Prepare Inputs & Target panel). You cannot generate a Derive node if the target is not a range, or if the Box-Cox rescaling is not selected.

The Derive node is created in Multiple mode and uses `@FIELD` in the expression so you can add the transformed target if required. For example using the following details:

- Target field name: `response`
- Transformed target field name: `response_transformed`
- Score field name: `$XR-response_transformed`

The Derive node would create a new field: `$XR-response_transformed_inverse`.

Note: If you are not using an automodeler or Ensemble node, you will need to edit the Derive node to transform the correct score field for your model.

Normalized continuous targets

By default, if you select the Rescale a continuous target with a Box-Cox transformation check box on the Prepare Inputs & Target panel this transforms the target and you create a new field that will be the target for your model building. For example if your original target was `response`, the new target will be `response_transformed`; models downstream of the ADP node will pick this new target up automatically.

However, this may cause issues, depending on the original target. For example, if the target was `Age`, the values of the new target will not be `Years`, but a transformed version of `Years`. This means you cannot look at the scores and interpret them since they aren't in recognizable units. In this case you can apply an inverse transformation that will turn your transformed units back into whatever they were meant to be. To do this:

1. After clicking Analyze Data to run the ADP analysis, select *Derive Node* from the *Generate* menu.
2. Place the Derive node after your nugget on the model canvas.

The Derive node will restore the score field to the original dimensions so that the prediction will be in the original `Years` values.

By default the Derive node transforms the score field generated by an auto-modeler or an ensembled model. If you are building an individual model, you need to edit the Derive node to derive from your actual score field. If you want to evaluate your model, you should add the transformed target to the Derive From field in the Derive node. This applies the same inverse transformation to the target and any downstream Evaluation or Analysis node will use the transformed data correctly as long as you switch those nodes to use field names instead of metadata.

If you also want to restore the original name, you can use a Filter node to remove the original target field if it's still there, and rename the target and score fields.

Type Node

Field properties can be specified in a source node or in a separate Type node. The functionality is similar in both nodes. The following properties are available:

- Field Double-click any field name to specify value and field labels for data in IBM® SPSS® Modeler. For example, field metadata imported from IBM SPSS Statistics can be viewed or modified here. Similarly, you can create new labels for fields and their values. The labels that you specify here are displayed throughout IBM SPSS Modeler depending on the selections you make in the stream properties dialog box.
- Measurement This is the measurement level, used to describe characteristics of the data in a given field. If all of the details of a field are known, it is called **fully instantiated**. For more information, see [Measurement levels](#).
Note: The measurement level of a field is different from its storage type, which indicates whether the data are stored as strings, integers, real numbers, dates, times, timestamps, or lists.
- Values This column enables you to specify options for reading data values from the data set, or use the Specify option to specify measurement levels and values in a separate dialog box. You can also choose to pass fields without reading their values. For more information, see [Data Values](#).
Note: You cannot amend the cell in this column if the corresponding Field entry contains a list.
- Missing Used to specify how missing values for the field will be handled. For more information, see [Defining Missing Values](#).
Note: You cannot amend the cell in this column if the corresponding Field entry contains a list.
- Check In this column, you can set options to ensure that field values conform to the specified values or ranges. For more information, see [Checking Type Values](#).
Note: You cannot amend the cell in this column if the corresponding Field entry contains a list.
- Role Used to tell modeling nodes whether fields will be Input (predictor fields) or Target (predicted fields) for a machine-learning process. Both and None are also available roles, along with Partition, which indicates a field used to partition records into separate samples for training, testing, and validation. The value Split specifies that separate models will be built for each possible value of the field. For more information, see [Setting the field role](#).

Several other options can be specified using the Type node window:

- Using the tools menu button, you can choose to Ignore Unique Fields once a Type node has been instantiated (either through your specifications, reading values, or running the stream). Ignoring unique fields will automatically ignore fields with only one value.
- Using the tools menu button, you can choose to Ignore Large Sets once a Type node has been instantiated. Ignoring large sets will automatically ignore sets with a large number of members.
- Using the tools menu button, you can choose to Convert Continuous Integers To Ordinal once a Type node has been instantiated. See the topic [Converting Continuous Data](#) for more information.
- Using the tools menu button, you can generate a Filter node to discard selected fields.
- Using the sunglasses toggle buttons, you can set the default for all fields to Read or Pass. The Types tab in the source node passes fields by default, while the Type node itself reads values by default.
- Using the Clear Values button, you can clear changes to field values made in this node (non-inherited values) and reread values from upstream operations. This option is useful for resetting changes that you may have made for specific fields upstream.
- Using the Clear All Values button, you can reset values for **all** fields read into the node. This option effectively sets the Values column to **Read** for all fields. This option is useful to reset values for all fields and reread values and types from upstream operations.
- Using the context menu, you can choose to Copy attributes from one field to another. See the topic [Copying Type Attributes](#) for more information.
- Using the View unused field settings option, you can view type settings for fields that are no longer present in the data or were once connected to this Type node. This is useful when reusing a Type node for datasets that have changed.
- [Viewing and setting information about types](#)
- [Measurement levels](#)
- [Converting Continuous Data](#)
- [What is instantiation?](#)
- [Data Values](#)
- [Defining Missing Values](#)
- [Checking Type Values](#)
- [Setting the field role](#)
- [Copying Type Attributes](#)
- [Field Format Settings Tab](#)

Related information

- [Field Operations Overview](#)
- [Viewing and setting information about types](#)
- [Measurement levels](#)
- [Converting Continuous Data](#)
- [What is instantiation?](#)
- [Data Values](#)
- [Checking Type Values](#)
- [Setting the field role](#)

Viewing and setting information about types

From various source nodes as well as the Type node, you can specify field metadata and properties that are invaluable to modeling and other work in IBM® SPSS® Modeler. These properties include:

- Specifying a usage type, such as range, set, ordered set, or flag, for each field in your dataset.
- Setting options for handling missing values and system nulls.
- Setting the role of a field for modeling purposes.
- Specifying values for a field as well as options used to automatically read values from the dataset.
- Specifying field and value labels.

Select help specific to your situation from the list below.

Measurement levels

Measurement level (formerly known as "data type" or "usage type") describes the usage of the data fields in IBM® SPSS® Modeler. The measurement level can be specified on the Types tab of a source or Type node. For example, you may want to set the measurement level for an integer field with values of 1 and 0 to *Flag*. This usually indicates that 1 = *True* and 0 = *False*.

Storage versus measurement. Note that the measurement level of a field is different from its storage type, which indicates whether data are stored as a string, integer, real number, date, time, or timestamp. While data types can be modified at any point in a stream using a Type node, storage must be determined at the source when reading data into IBM SPSS Modeler (although it can subsequently be changed using a conversion function). See the topic [Setting Field Storage and Formatting](#) for more information.

Some modeling nodes indicate the permitted measurement level types for their input and target fields by means of icons on their Fields tab.

Measurement level icons

Table 1. Measurement level icons

Icon	Measurement level
	Default
	Continuous
	Categorical
	Flag
	Nominal
	Ordinal
	Typeless
	Collection
	Geospatial

The following measurement levels are available:

- Default Data whose storage type and values are unknown (for example, because they have not yet been read) are displayed as <Default>.
- Continuous Used to describe numeric values, such as a range of 0–100 or 0.75–1.25. A continuous value can be an integer, real number, or date/time.
- Categorical Used for string values when an exact number of distinct values is unknown. This is an **uninstantiated** data type, meaning that all possible information about the storage and usage of the data is not yet known. Once data have been read, the measurement level will be *Flag*, *Nominal*, or *Typeless*, depending on the maximum number of members for nominal fields specified in the Stream Properties dialog box.
- Flag Used for data with two distinct values that indicate the presence or absence of a trait, such as `true` and `false`, `Yes` and `No` or 0 and 1. The values used may vary, but one must always be designated as the "true" value, and the other as the "false" value. Data may be represented as text, integer, real number, date, time, or timestamp.
- Nominal Used to describe data with multiple distinct values, each treated as a member of a set, such as `small/medium/large`. Nominal data can have any storage—numeric, string, or date/time. Note that setting the measurement level to *Nominal* does not automatically change the values to string storage.
- Ordinal Used to describe data with multiple distinct values that have an inherent order. For example, salary categories or satisfaction rankings can be typed as ordinal data. The order is defined by the natural sort order of the data elements. For example, 1, 3, 5 is the default sort order for a set of integers, while **HIGH**, **LOW**, **NORMAL** (ascending alphabetically) is the order for a set of strings. The ordinal measurement level enables you to define a set of categorical data as ordinal data for the purposes of visualization, model building, and export to other applications (such as IBM SPSS Statistics) that recognize ordinal data as a distinct type. You can use an ordinal field anywhere that a nominal field can be used. Additionally, fields of any storage type (real, integer, string, date, time, and so on) can be defined as ordinal.
- Typeless Used for data that does not conform to any of the above types, for fields with a single value, or for nominal data where the set has more members than the defined maximum. It is also useful for cases in which the measurement level would otherwise be a set with many members (such as an account number). When you select Typeless for a field, the role is automatically set to None, with Record ID as the only alternative. The default maximum size for sets is 250 unique values. This number can be adjusted or disabled on the Options tab of the Stream Properties dialog box, which can be accessed from the Tools menu.

- Collection Used to identify non-geospatial data that is recorded in a list. A collection is effectively a list field of zero depth, where the elements in that list have one of the other measurement levels.
For more information about lists, see [List storage and associated measurement levels](#).
- Geospatial Used with the List storage type to identify geospatial data. Lists can be either List of Integer or List of Real fields with a list depth that is between zero and two, inclusive.
For more information, see [Geospatial measurement sublevels](#).

You can manually specify measurement levels, or you can allow the software to read the data and determine the measurement level based on the values that it reads.

Alternatively, where you have several continuous data fields that should be treated as categorical data, you can choose an option to convert them. See the topic [Converting Continuous Data](#) for more information.

To use auto-typing

1. In either a Type node or the Types tab of a source node, set the Values column to <Read> for the desired fields. This will make metadata available to all nodes downstream. You can quickly set all fields to <Read> or <Pass> using the sunglasses buttons on the dialog box.
2. Click Read Values to read values from the data source immediately.

To manually set the measurement level for a field

1. Select a field in the table.
 2. From the drop-down list in the Measurement column, select a measurement level for the field.
 3. Alternatively, you can use Ctrl-A or Ctrl-click to select multiple fields before using the drop-down list to select a measurement level.
- [Geospatial measurement sublevels](#)

Geospatial measurement sublevels

The Geospatial measurement level, which is used with the List storage type, has six sublevels that are used to identify different types of geospatial data.

- Point - Identifies a specific location; for example, the center of a city.
- Polygon - A series of points that identifies the single boundary of a region and its location; for example, a county.
- LineString - Also referred to as a Polyline or just a Line, a LineString is a series of points that identifies the route of a line. For example, a LineString might be a fixed item, such as road, river, or railway; or the track of something that moves, such as an aircraft's flight path or a ship's voyage.
- MultiPoint - Used when each row in your data contains multiple points per region. For example, if each row represents a city street, the multiple points for each street can be used to identify every street lamp.
- MultiPolygon - Used when each row in your data contains several polygons. For example, if each row represents the outline of a country, the US can be recorded as several polygons to identify the different areas such as the mainland, Alaska, and Hawaii.
- MultiLineString - Used when each row in your data contains several lines. Because lines cannot branch you can use a MultiLineString to identify a group of lines. For example, data such as the navigable waterways or the railway network in each country.

These measurement sublevels are used with the List storage type. For more information, see [List storage and associated measurement levels](#).

Restrictions

You must be aware of some restrictions when you use geospatial data.

- The coordinate system can affect the format of the data. For example, a Projected coordinate system uses the coordinate values x, y, and (when required) z, whereas a Geographic coordinate system uses the coordinate values longitude, latitude, and (when required) a value for either altitude or depth.
For more information about coordinate systems, see [Setting geospatial options for streams](#).
- A LineString cannot cross over itself.
- A Polygon is not self-closing; for each Polygon you must ensure that the first and last points are defined as the same.
- The direction of the data in a MultiPolygon is important; clockwise indicates a solid form and counterclockwise indicates a hole. For example, if you record an area of a country that contains lakes; the main land area border can be recorded in a clockwise direction and the shape of each lake in a counterclockwise direction.
- A Polygon cannot intersect with itself. An example of this intersection would be if you tried to plot the boundary of the polygon as a continuous line in the form of the figure 8.
- MultiPolygons cannot overlap each other.
- For Geospatial fields, the only relevant storage types are Real and Integer (the default setting is Real).

Geospatial measurement sublevel icons

Table 1. Geospatial measurement sublevel icons

Icon	Measurement level
	Point
	Polygon
	LineString
	MultiPoint
	MultiPolygon
	MultiLineString

Converting Continuous Data

Treating categorical data as continuous can have a serious effect on the quality of a model, especially if it's the target field; for example, producing a regression model rather than a binary model. To prevent this you can convert integer ranges to categorical types such as *Ordinal* or *Flag*.

1. From the Operations and Generate menu button (with the tool symbol), select Convert Continuous Integers To Ordinal. The conversion values dialog is displayed.
2. Specify the size of range that will be automatically converted; this applies to any range up to and including the size you enter.
3. Click OK. The affected ranges are converted to either *Flag* or *Ordinal* and displayed on the Types tab of the Type node.

Results of the Conversion

- Where a *Continuous* field with integer storage is changed to *Ordinal*, the lower and upper values are expanded to include all of the integer values from the lower value to the upper value. For example, if the range is 1, 5, the set of values is 1, 2, 3, 4, 5.
- If the *Continuous* field changes to *Flag*, the lower and upper values become the false and true values of the flag field.

Related information

- [Type Node](#)
- [Viewing and setting information about types](#)
- [Measurement levels](#)
- [What is instantiation?](#)
- [Data Values](#)
- [Checking Type Values](#)
- [Setting the field role](#)

What is instantiation?

Instantiation is the process of reading or specifying information, such as storage type and values for a data field. To optimize system resources, instantiating is a user-directed process—you tell the software to read values by specifying options on the Types tab in a source node or by running data through a Type node.

- Data with unknown types are also referred to as *uninstantiated*. Data whose storage type and values are unknown are displayed in the *Measurement* column of the Types tab as <Default>.
- When you have some information about a field's storage, such as string or numeric, the data are called *partially instantiated*. Categorical or Continuous are partially instantiated measurement levels. For example, Categorical specifies that the field is symbolic, but you don't know whether it is nominal, ordinal, or flag.
- When all of the details about a type are known, including the values, a *fully instantiated* measurement level—nominal, ordinal, flag, or continuous—is displayed in this column. Note that the *continuous* type is used for both partially instantiated and fully instantiated data fields. Continuous data can be either integers or real numbers.

During the execution of a data stream with a Type node, uninstantiated types immediately become partially instantiated, based on the initial data values. Once all of the data have passed through the node, all data become fully instantiated unless values were set to <Pass>. If execution is interrupted, the data will remain partially instantiated. Once the Types tab has been instantiated, the values of a field are static at that point in the stream. This means that any upstream changes will not affect the values of a particular field, even if you rerun the stream. To change or update the values based on new data or added manipulations, you need to edit them in the Types tab itself or set the value for a field to <Read> or <Read +>.

When to instantiate

Generally, if your dataset is not very large and you do not plan to add fields later in the stream, instantiating at the source node is the most convenient method. However, instantiating in a separate Type node is useful when:

- The dataset is large, and the stream filters a subset prior to the Type node.
- Data have been filtered in the stream.
- Data have been merged or appended in the stream.
- New data fields are derived during processing.

Note: If you export data in a database export node, the data must be fully instantiated.

Data Values

Using the Values column of the Types tab, you can read values automatically from the data, or you can specify measurement levels and values in a separate dialog box.

The options available from the Values drop-down list provide instructions for auto-typing, as shown in the following table.

Table 1. Instructions for auto-typing

Option	Function
<Read>	Data is read when the node is executed.
<Read+>	Data is read and appended to the current data (if any exist).
<Pass>	No data is read.
<Current>	Keep current data values.
Specify...	A separate dialog box opens for you to specify values and measurement level options.

Executing a Type node or clicking Read Values auto-types and reads values from your data source based on your selection. These values can also be specified manually using the Specify option or by double-clicking a cell in the Field column.

After you make changes for fields in the Type node, you can reset value information using the following buttons on the dialog box toolbar:

- Using the Clear All Values button, you can clear changes to field values made in this node (non-inherited values) and reread values from upstream operations. This option is useful for resetting changes that you may have made for specific fields upstream.
- Using the Clear Values button, you can reset values for **all** fields read into the node. This option effectively sets the *Values* column to **Read** for all fields. This option is useful to reset values for all fields and reread values and measurement levels from upstream operations.

Grey text in the Values column

Within either a Type node, or a Source node, if the data in the Values column is shown in black text, it indicates that the values of that field have been read and are stored within that node. If no black text is present in this field, the values of that field have not been read and are determined further upstream.

There are occasions when you can see the data as gray text. This occurs when SPSS® Modeler can identify or infer the valid values of a field without actually reading and storing the data. This is likely to occur if you use one of the following nodes:

- User Input node. Because the data is defined within the node, the range of values for a field is always known, even if the values have not been stored in the node.
- Statistics File source node. If there is metadata present for the data types it enables SPSS Modeler to infer the possible range of values without reading or storing the data.

In either node, the values are shown in gray text until you click Read Values.

Warning: If you do not instantiate the data in your stream, and your data values are shown in gray, any checking of type values that you set in the Check column is not applied.

- [Using the Values Dialog Box](#)
- [Specifying Values and Labels for Continuous Data](#)
- [Specifying Values and Labels for Nominal and Ordinal Data](#)
- [Specifying Values for a Flag](#)
- [Specifying Values for Collection Data](#)
- [Specifying values for geospatial data](#)

Related information

- [Measurement levels](#)

- [Using the Values Dialog Box](#)
 - [Copying Type Attributes](#)
 - [Type Node](#)
 - [Viewing and setting information about types](#)
 - [Converting Continuous Data](#)
 - [What is instantiation?](#)
 - [Checking Type Values](#)
 - [Setting the field role](#)
-

Using the Values Dialog Box

Clicking the Values or Missing column of the Types tab displays a drop-down list of predefined values. Choosing the Specify... option on this list opens a separate dialog box where you can set options for reading, specifying, labeling, and handling values for the selected field.

Many of the controls are common to all types of data. These common controls are discussed here.

Measurement Displays the currently selected measurement level. You can change the setting to reflect the way that you intend to use data. For instance, if a field called `day_of_week` contains numbers that represent individual days, you might want to change this to nominal data in order to create a distribution node that examines each category individually.

Storage Displays the storage type if known. Storage types are unaffected by the measurement level that you choose. To alter the storage type, you can use the Data tab in Fixed File and Variable File source nodes, or a conversion function in a Filler node.

Model Field For fields generated as a result of scoring a model nugget, model field details can also be viewed. These include the name of the target field as well as the role of the field in modeling (whether a predicted value, probability, propensity, and so on).

Values Select a method to determine values for the selected field. Selections that you make here override any selections that you made earlier from the Values column of the Type node dialog box. Choices for reading values include the following:

- Read from data Select to read values when the node is executed. This option is the same as <Read>.
- Pass Select not to read data for the current field. This option is the same as <Pass>.
- Specify values and labels Options here are used to specify values and labels for the selected field. Used with value checking, use this option to specify values that are based on your knowledge of the current field. This option activates unique controls for each type of field. Options for values and labels are covered individually in subsequent topics.
Note: You cannot specify values or labels for a field whose measurement level is Typeless or <Default>.
- Extend values from data Select to append the current data with the values that you enter here. For example, if `field_1` has a range from (0,10), and you enter a range of values from (8,16), the range is extended by adding the 16, without removing the original minimum. The new range would be (0,16). Choosing this option automatically sets the auto-typing option to <Read+>.

Max list length Only available for data with a measurement level of either Geospatial or Collection. Set the maximum length of the list by specifying the number of elements the list can contain.

Max string length Only available for typeless data; use this field when you are generating SQL to create a table. Enter the value of the largest string in your data; this generates a column in the table that is big enough for the string. If the string length value is not available, a default string size is used that may not be appropriate for the data (for example, if the value is too small, errors can occur when writing data to the table; too large a value could adversely affect performance).

Check values Select a method of coercing values to conform to the specified continuous, flag, or nominal values. This option corresponds to the Check column in the Type node dialog box, and settings made here override those in the dialog box. Used with the Specify values and labels option, value checking allows you to conform values in the data with expected values. For example, if you specify values as 1, 0 and then use the Discard option, you can discard all records with values other than 1 or 0.

Define blanks Select to activate the following controls that you use to declare missing values or blanks in your data.

- Missing values Use this table to define specific values (such as 99 or 0) as blanks. The value should be appropriate for the storage type of the field.
- Range Used to specify a range of missing values, for example, ages 1–17 or greater than 65. If a bound value is left blank, then the range is unbounded; for example, if a lower bound of 100 is specified with no upper bound, then all values greater than or equal to 100 is defined as missing. The bound values are inclusive; for example, a range with a lower bound of 5 and an upper bound of 10 includes 5 and 10 in the range definition. A missing value range can be defined for any storage type, including date/time and string (in which case the alphabetic sort order is used to determine whether a value is within the range).
- Null/White space You can also specify system nulls (displayed in the data as `$null$`) and white space (string values with no visible characters) as blanks.
Note: The Type node also treats empty strings as white space for purposes of analysis, although they are stored differently internally and may be handled differently in certain cases.

Note: To code blanks as undefined or `$null$`, use the Filler node.

Description Use this text box to specify a field label. These labels appear in various locations, such as in graphs, tables, output, and model browsers, depending on selections you make in the Stream Properties dialog box.

Related information

- [Type Node](#)
 - [Specifying Values and Labels for Nominal and Ordinal Data](#)
 - [Specifying Values and Labels for Continuous Data](#)
 - [Specifying Values for a Flag](#)
-

Specifying Values and Labels for Continuous Data

The *Continuous* measurement level is used for numeric fields. There are three storage types for continuous data:

- Real
- Integer
- Date/Time

The same dialog box is used to edit all continuous fields; the storage type is displayed for reference only.

Specifying Values

The following controls are unique to continuous fields and are used to specify a range of values:

Lower. Specify a lower limit for the value range.

Upper. Specify an upper limit for the value range.

Specifying Labels

You can specify labels for any value of a range field. Click the Labels button to open a separate dialog box for specifying value labels.

- [Values and Labels Subdialog Box](#)

Related information

- [Type Node](#)
 - [Using the Values Dialog Box](#)
-

Values and Labels Subdialog Box

Clicking Labels in the Values dialog box for a range field opens a new dialog box in which you can specify labels for any value in the range.

You can use the *Values* and *Labels* columns in this table to define value and label pairs. Currently defined pairs are shown here. You can add new label pairs by clicking in an empty cell and entering a value and its corresponding label. *Note:* Adding value/value-label pairs to this table will not cause any new values to be added to the field. Instead, it simply creates metadata for the field value.

The labels that you specify in the Type node are displayed in many places (as ToolTips, output labels, and so on), depending on selections that you make in the stream properties dialog box.

Specifying Values and Labels for Nominal and Ordinal Data

Nominal (set) and ordinal (ordered set) measurement levels indicate that the data values are used discretely as a member of the set. The storage types for a set can be string, integer, real number, or date/time.

The following controls are unique to nominal and ordinal fields and are used to specify values and labels:

Values. The *Values* column in the table allows you to specify values based on your knowledge of the current field. Using this table, you can enter expected values for the field and check the dataset's conformity to these values using the Check Values drop-down list. Using the arrow and delete buttons, you can modify existing values as well as reorder or delete values.

Labels. The *Labels* column enables you to specify labels for each value in the set. These labels appear in a variety of locations, such as graphs, tables, output, and model browsers, depending on selections that you make in the stream properties dialog box.

Related information

- [Type Node](#)
 - [Using the Values Dialog Box](#)
-

Specifying Values for a Flag

Flag fields are used to display data that have two distinct values. The storage types for flags can be string, integer, real number, or date/time.

True. Specify a flag value for the field when the condition is met.

False. Specify a flag value for the field when the condition is not met.

Labels. Specify labels for each value in the flag field. These labels appear in a variety of locations, such as graphs, tables, output, and model browsers, depending on selections that you make in the stream properties dialog box.

Related information

- [Type Node](#)
 - [Using the Values Dialog Box](#)
-

Specifying Values for Collection Data

Collection fields are used to display non-geospatial data that is in a list.

The only item that you can set for the Collection Measurement level is the List measure. By default this measure is set to Typeless but you can select another value from to set the measurement level of the elements within the list. You can choose one of the following options:

- Typeless
- Continuous
- Nominal
- Ordinal
- Flag

Related information

- [Type Node](#)
 - [Using the Values Dialog Box](#)
-

Specifying values for geospatial data

Geospatial fields are used to display geospatial data that is in a list.

For the Geospatial Measurement level, you can set the following options to set the measurement level of the elements within the list:

Type Select the measurement sublevel of the geospatial field. The available sublevels are determined by the depth of the list field; the defaults are: Point (zero depth), LineString (depth of one), and Polygon (depth of one).

For more information about sublevels, see [Geospatial measurement sublevels](#).

For more information about list depths, see [List storage and associated measurement levels](#).

Coordinate system This option is only available if you changed the measurement level to geospatial from a non-geospatial level. To apply a coordinate system to your geospatial data, select this check box. By default, the coordinate system set in the Tools > Stream Properties > Options > Geospatial pane is shown. To use a different coordinate system, click the Change button to display the Select Coordinate System dialog box and choose the system that you require.

For more information about coordinate systems, see [Setting geospatial options for streams](#).

Defining Missing Values

The Missing column of the Types tab indicates whether missing value handling has been defined for a field. The possible settings are:

On (*). Indicates that missing values handling is defined for this field. This could be done by means of a downstream Filler node, or through an explicit specification using the Specify option (see below).

Off. The field has no missing value handling defined.

Specify. Choose this option to display a dialog where you can declare explicit values to be considered as missing values for this field.

Checking Type Values

Turning on the Check option for each field examines all values in that field to determine whether they comply with the current type settings or the values that you have specified in the Specify Values dialog box. This is useful for cleaning up datasets and reducing the size of a dataset within a single operation.

The setting of the Check column in the Type node dialog box determines what happens when a value outside of the type limits is discovered. To change the Check settings for a field, use the drop-down list for that field in the Check column. To set the Check settings for all fields, click in the Field column and press Ctrl-A. Then use the drop-down list for any field in the Check column.

The following Check settings are available:

None. Values will be passed through without checking. This is the default setting.

Nullify. Change values outside of the limits to the system null (\$null\$).

Coerce. Fields whose measurement levels are fully instantiated will be checked for values that fall outside the specified ranges. Unspecified values will be converted to a legal value for that measurement level using the following rules:

- For flags, any value other than the true and false value is converted to the false value.
- For sets (nominal or ordinal), any unknown value is converted to the first member of the set's values.
- Numbers greater than the upper limit of a range are replaced by the upper limit.
- Numbers less than the lower limit of a range are replaced by the lower limit.
- Null values in a range are given the midpoint value for that range.

Discard. When illegal values are found, the entire record is discarded.

Warn. The number of illegal items is counted and reported in the stream properties dialog box when all of the data have been read.

Abort. The first illegal value encountered terminates the running of the stream. The error is reported in the stream properties dialog box.

Related information

- [Type Node](#)
- [Viewing and setting information about types](#)
- [Measurement levels](#)
- [Converting Continuous Data](#)
- [What is instantiation?](#)
- [Data Values](#)
- [Setting the field role](#)
- [Copying Type Attributes](#)

Setting the field role

The role of a field specifies how it is used in model building—for example, whether a field is an input or target (the thing being predicted).

Note: The Partition, Frequency and Record ID roles can each be applied to a single field only.

The following roles are available:

Input. The field will be used as an input to machine learning (a predictor field).

Target. The field will be used as an output or target for machine learning (one of the fields that the model will try to predict).

Both. The field will be used as both an input and an output by the Apriori node. All other modeling nodes will ignore the field.

None. The field will be ignored by machine learning. Fields whose measurement level has been set to Typeless are automatically set to None in the Role column.

Partition. Indicates a field used to partition the data into separate samples for training, testing, and (optional) validation purposes. The field must be an instantiated set type with two or three possible values (as defined in the Field Values dialog box). The first value represents the training sample, the second represents the testing sample, and the third (if present) represents the validation sample. Any additional values are ignored, and flag fields cannot be used. Note that to use the partition in an analysis, partitioning must be enabled on the Model Options tab in the appropriate model-building or analysis node. Records with null values for the partition field are excluded from the analysis when partitioning is enabled. If multiple partition fields have been defined in the stream, a single partition field must be specified on the Fields tab in each applicable modeling node. If a suitable field doesn't already exist in your data, you can create one using a Partition node or Derive node. See the topic [Partition Node](#) for more information.

Split. (Nominal, ordinal and flag fields only) Specifies that a model is to be built for each possible value of the field.

Frequency. (Numeric fields only) Setting this role enables the field value to be used as a frequency weighting factor for the record. This feature is supported by C&R Tree, CHAID, QUEST and Linear models only; all other nodes ignore this role. Frequency weighting is enabled by means of the Use frequency weight option on the Fields tab of those modeling nodes that support the feature.

Record ID. The field will be used as the unique record identifier. This feature is ignored by most nodes; however it is supported by Linear models, and is required for the IBM Netezza in-database mining nodes.

Copying Type Attributes

You can easily copy the attributes of a type, such as values, checking options, and missing values from one field to another:

1. Right-click on the field whose attributes you want to copy.
2. From the context menu, choose Copy.
3. Right-click on the field(s) whose attributes you want to change.
4. From the context menu, choose Paste Special. *Note:* You can select multiple fields using the Ctrl-click method or by using the Select Fields option from the context menu.

A new dialog box opens, from which you can select the specific attributes that you want to paste. If you are pasting into multiple fields, the options that you select here will apply to all target fields.

Paste the following attributes. Select from the list below to paste attributes from one field to another.

- **Type.** Select to paste the measurement level.
- **Values.** Select to paste the field values.
- **Missing.** Select to paste missing value settings.
- **Check.** Select to paste value checking options.
- **Role.** Select to paste the role of a field.

Related information

- [Type Node](#)

Field Format Settings Tab

The Format tab on the Table and Type nodes shows a list of current or unused fields and formatting options for each field. Following is a description of each column in the field formatting table:

Field. This shows the name of the selected field.

Format. By double-clicking a cell in this column, you can specify formatting for fields on an individual basis using the dialog box that opens. See the topic [Setting Field Format options](#) for more information. Formatting specified here overrides formatting specified in the overall stream properties.

Note: The Statistics Export and Statistics Output nodes export .sav files that include per-field formatting in their metadata. If a per-field format is specified that is not supported by the IBM® SPSS® Statistics .sav file format, then the node will use the IBM SPSS Statistics default format.

Justify. Use this column to specify how the values should be justified within the table column. The default setting is Auto, which left-justifies symbolic values and right-justifies numeric values. You can override the default by selecting Left, Right, or Center.

Column Width. By default, column widths are automatically calculated based on the values of the field. To override the automatic width calculation, click a table cell and use the drop-down list to select a new width. To enter a custom width not listed here, open the Field Formats subdialog box by double-clicking a table cell in the Field or Format column. Alternatively, you can right-click on a cell and choose Set Format.

View Current Fields. By default, the dialog box shows the list of currently active fields. To view the list of unused fields, select View unused fields settings.

Context Menu. The context menu for this tab provides various selection and setting update options. Right-click in a column to display this menu.

- **Select All.** Selects all fields.
 - **Select None.** Clears the selection.
 - **Select Fields.** Selects fields based on type or storage characteristics. Options are Select Categorical, Select Continuous (numeric), Select Typeless, Select Strings, Select Numbers, or Select Date/Time. See the topic [Measurement levels](#) for more information.
 - **Set Format.** Opens a subdialog box for specifying date, time, and decimal options on a per-field basis.
 - **Set Justify.** Sets the justification for the selected field(s). Options are Auto, Center, Left, or Right.
 - **Set Column Width.** Sets the field width for selected fields. Specify Auto to read width from the data. Or you can set field width to 5, 10, 20, 30, 50, 100, or 200.
 - [**Setting Field Format options**](#)
-

Setting Field Format options

Field formatting is specified on a subdialog box available from the Format tab on the Type and Table nodes. If you have selected more than one field before opening this dialog box, then settings from the first field in the selection are used for all. Clicking OK after making specifications here will apply these settings to all fields selected on the Format tab.

The following options are available on a per-field basis. Many of these settings can also be specified in the stream properties dialog box. Any settings made at the field level override the default specified for the stream.

Date format. Select a date format to be used for date storage fields or when strings are interpreted as dates by CLEM date functions.

Time format. Select a time format to be used for time storage fields or when strings are interpreted as times by CLEM time functions.

Number display format. You can choose from standard (####.###), scientific (#.###E+##), or currency display formats (\$###.##).

Decimal symbol. Select either a comma (,) or period (.) as a decimal separator.

Grouping symbol. For number display formats, select the symbol used to group values (For example, the comma in 3,000.00). Options include none, period, comma, space, and locale-defined (in which case the default for the current locale is used).

Decimal places (standard, scientific, currency, export). For number display formats, specifies the number of decimal places to be used when displaying real numbers. This option is specified separately for each display format. Note that the Export decimal places setting only applies to flat file exports, and it overrides the stream properties. The stream default for flat file export is whatever is specified for the Standard decimal places setting in the stream properties. The number of decimal places exported by the XML Export node is always 6.

Justify. Specifies how the values should be justified within the column. The default setting is Auto, which left-justifies symbolic values and right-justifies numeric values. You can override the default by selecting left, right, or center.

Column width. By default, column widths are automatically calculated based on the values of the field. You can specify a custom width in intervals of five using the arrows to the right of the list box.

Filtering or renaming fields

You can rename or exclude fields at any point in a stream. For example, as a medical researcher, you may not be concerned about the potassium level (field-level data) of patients (record-level data); therefore, you can filter out the K (potassium) field. This can be done using a separate Filter node or using the Filter tab on a source or output node. The functionality is the same regardless of which node it is accessed from.

- From source nodes, such as Variable File, Fixed File, Statistics File, XML, or Extension Import, you can rename or filter fields as the data are read into IBM® SPSS® Modeler.
 - Using a Filter node, you can rename or filter fields at any point in the stream.
 - From Statistics Export, Statistics Transform, Statistics Model, and Statistics Output nodes, you can filter or rename fields to conform to IBM SPSS Statistics naming standards. See the topic [Renaming or Filtering Fields for IBM SPSS Statistics](#) for more information.
 - You can use the Filter tab in any of the above nodes to define or edit multiple response sets. See the topic [Editing Multiple Response Sets](#) for more information.
 - Finally, you can use a Filter node to map fields from one source node to another.
 - [**Setting filtering options**](#)
-

Setting filtering options

The table used on the Filter tab shows the name of each field as it comes into the node as well as the name of each field as it leaves. You can use the options in this table to rename or filter out fields that are duplicates or are unnecessary for downstream operations.

- Field. Displays the input fields from currently connected data sources.
- Filter. Displays the filter status of all input fields. Filtered fields include a red X in this column, indicating that this field will not be passed downstream. Click in the *Filter* column for a selected field to turn filtering on and off. You can also select options for multiple fields simultaneously using the Shift-click method of selection.
- Field. Displays the fields as they leave the Filter node. Duplicate names are displayed in red. You can edit field names by clicking in this column and entering a new name. Or, remove fields by clicking in the *Filter* column to disable duplicate fields.

All columns in the table can be sorted by clicking on the column header.

View current fields. Select to view fields for datasets actively connected to the Filter node. This option is selected by default and is the most common method of using Filter nodes.

View unused field settings. Select to view fields for datasets that were once but are no longer connected to the Filter node. This option is useful when copying Filter nodes from one stream to another or when saving and reloading Filter nodes.

Filter Button Menu

Click the Filter button in the upper left corner of the dialog box to access a menu that provides a number of shortcuts and other options.

You can choose to:

- Remove all fields.
- Include all fields.
- Toggle all fields.
- Remove duplicates. Note that selecting this option removes all occurrences of the duplicate name, including the first one.
- Rename fields and multiple response sets to conform with other applications. See the topic [Renaming or Filtering Fields for IBM SPSS Statistics](#) for more information.
- Truncate field names.
- Anonymize field and multiple response set names.
- Use input field names.
- Edit Multiple Response Sets. See the topic [Editing Multiple Response Sets](#) for more information.
- Set the default filter state.

You can also use the arrow toggle buttons at the top of the dialog box to specify whether you want to include or discard fields by default. This is useful for large datasets where only a few fields are to be included downstream. For example, you can select only the fields you want to keep and specify that all others should be discarded (rather than individually selecting all of the fields to discard).

- [Truncating Field Names](#)
- [Anonymizing Field Names](#)
- [Editing Multiple Response Sets](#)

Truncating Field Names

From the Filter button menu (upper left corner of the Filter tab), you can choose to truncate field names.

Maximum length. Specify a number of characters to limit the length of field names.

Number of digits. If field names, when truncated, are no longer unique, they will be further truncated and differentiated by adding digits to the name. You can specify the number of digits used. Use the arrow buttons to adjust the number.

For example, the following table illustrates how field names in a medical dataset are truncated using the default settings (maximum length=8 and number of digits=2).

Table 1. Field name truncation

Field Names	Truncated Field Names
Patient Input 1	Patien01
Patient Input 2	Patien02
Heart Rate	HeartRat
BP	BP

Related information

- [Filtering or renaming fields](#)

Anonymizing Field Names

You can anonymize field names from any node that includes a Filter tab by clicking the Filter button menu in the upper left corner and choosing Anonymize Field Names. Anonymized field names consist of a string prefix plus a unique numeric-based value.

Anonymize names of. Choose Selected fields only to anonymize only the names of fields already selected on the Filter tab. The default is All fields, which anonymizes all field names.

Field names prefix. The default prefix for anonymized field names is anon_ ; choose Custom and type your own prefix if you want a different one.

Anonymize multiple response sets. Anonymizes the names of multiple response sets in the same manner as fields. See the topic [Editing Multiple Response Sets](#) for more information.

To restore the original field names, choose Use Input Field Names from the filter button menu.

Related information

- [Filtering or renaming fields](#)

Editing Multiple Response Sets

You can add or edit multiple response sets from any node that includes a Filter tab by clicking the Filter button menu in the upper left corner and choosing Edit Multiple Response Sets.

Multiple response sets are used to record data that can have more than one value for each case—for example, when asking survey respondents which museums they have visited or which magazines they read. Multiple response sets can be imported into IBM® SPSS® Modeler using a Data Collection source node or a Statistics File source node and can be defined in IBM SPSS Modeler using a Filter node.

Click New to create a new multiple response set, or click Edit to modify an existing set.

Name and label. Specifies the name and description for the set.

Type. Multiple response questions can be handled in one of two ways:

- **Multiple dichotomy set.** A separate flag field is created for each possible response; thus, if there are 10 magazines, there are 10 flag fields, each of which can have values, such as 0 or 1 for *true* or *false*. The counted value allows you to specify which value is counted as true. This method is useful when you want to allow respondents to choose all options that apply.
- **Multiple category set.** A nominal field is created for each response up to the maximum number of answers from a given respondent. Each nominal field has values representing the possible answers, such as 1 for *Time*, 2 for *Newsweek*, and 3 for *PC Week*. This method is most useful when you want to limit the number of answers—for example, when asking respondents to choose the three magazines they read most frequently.

Fields in set. Use the icons on the right to add or remove fields.

Comments

- All fields included in a multiple response set must have the same storage.
- Sets are distinct from the fields they include. For example, deleting a set will not cause the fields it includes to be deleted—merely the links between those fields. The set is still visible upstream from the point of deletion but is not visible downstream.
- If fields are renamed using a Filter node (directly on the tab or by choosing the Rename for IBM SPSS Statistics, Truncate, or Anonymize options on the Filter menu), any references to these fields used in multiple response sets will also be updated. However, any fields in a multiple response set that are dropped by the Filter node will not be removed from the multiple response set. Such fields, although no longer visible in the stream, are still referenced by the multiple response set; this could be a consideration when exporting, for example.

Related information

- [Filtering or renaming fields](#)

Derive node

One of the most powerful features in IBM® SPSS® Modeler is the ability to modify data values and derive new fields from existing data. During lengthy data mining projects, it is common to perform several derivations, such as extracting a customer ID from a string of Web log data or

creating a customer lifetime value based on transaction and demographic data. All of these transformations can be performed, using a variety of field operations nodes.

Several nodes provide the ability to derive new fields:

	The Derive node modifies data values or creates new fields from one or more existing fields. It creates fields of type formula, flag, nominal, state, count, and conditional.
	The Reclassify node transforms one set of categorical values to another. Reclassification is useful for collapsing categories or regrouping data for analysis.
	The Binning node automatically creates new nominal (set) fields based on the values of one or more existing continuous (numeric range) fields. For example, you can transform a continuous income field into a new categorical field containing groups of income as deviations from the mean. Once you have created bins for the new field, you can generate a Derive node based on the cut points.
	The Set to Flag node derives multiple flag fields based on the categorical values defined for one or more nominal fields.
	The Restructure node converts a nominal or flag field into a group of fields that can be populated with the values of yet another field. For example, given a field named <i>payment type</i> , with values of <i>credit</i> , <i>cash</i> , and <i>debit</i> , three new fields would be created (<i>credit</i> , <i>cash</i> , <i>debit</i>), each of which might contain the value of the actual payment made.
	The History node creates new fields containing data from fields in previous records. History nodes are most often used for sequential data, such as time series data. Before using a History node, you may want to sort the data using a Sort node.

Using the Derive node

Using the Derive node, you can create six types of new fields from one or more existing fields:

- Formula. The new field is the result of an arbitrary CLEM expression.
- Flag. The new field is a flag, representing a specified condition.
- Nominal. The new field is nominal, meaning that its members are a group of specified values.
- State. The new field is one of two states. Switching between these states is triggered by a specified condition.
- Count. The new field is based on the number of times that a condition has been true.
- Conditional. The new field is the value of one of two expressions, depending on the value of a condition.

Each of these nodes contains a set of special options in the Derive node dialog box. These options are discussed in subsequent topics.

Note that use of the following may change row order:

- Executing in a database via SQL pushback
 - Executing via remote IBM SPSS Analytic Server
 - Using functions that run in embedded IBM SPSS Analytic Server
 - Deriving a list (for example, see [Deriving a list or geospatial field](#))
 - Calling any of the spatial functions
- [Setting Basic Options for the Derive Node](#)
 - [Deriving Multiple Fields](#)
 - [Setting Derive Formula Options](#)
 - [Setting Derive Flag Options](#)
 - [Setting Derive Nominal Options](#)
 - [Setting Derive State Options](#)
 - [Setting Derive Count Options](#)
 - [Setting Derive Conditional Options](#)
 - [Recoding Values with the Derive Node](#)

Setting Basic Options for the Derive Node

At the top of the dialog box for Derive nodes are a number of options for selecting the type of Derive node that you need.

Mode. Select Single or Multiple, depending on whether you want to derive multiple fields. When Multiple is selected, the dialog box changes to include options for multiple Derive fields.

Derive field. For simple Derive nodes, specify the name of the field that you want to derive and add to each record. The default name is DeriveN, where N is the number of Derive nodes that you have created thus far during the current session.

Derive as. Select a type of Derive node, such as Formula or Nominal, from the drop-down list. For each type, a new field is created based on the conditions that you specify in the type-specific dialog box.

Selecting an option from the drop-down list will add a new set of controls to the main dialog box according to the properties of each Derive node type.

Field type. Select a measurement level, such as continuous, categorical, or flag, for the newly derived node. This option is common to all forms of Derive nodes.

Note: Deriving new fields often requires the use of special functions or mathematical expressions. To help you create these expressions, an Expression Builder is available from the dialog box for all types of Derive nodes and provides rule checking as well as a complete list of CLEM expressions.

Related information

- [Setting Derive Formula Options](#)
- [Setting Derive Flag Options](#)
- [Setting Derive Nominal Options](#)
- [Setting Derive Count Options](#)
- [Setting Derive State Options](#)
- [Setting Derive Conditional Options](#)
- [Derive node](#)
- [Deriving Multiple Fields](#)
- [Recoding Values with the Derive Node](#)

Deriving Multiple Fields

Setting the mode to Multiple within a Derive node gives you the capability to derive multiple fields based on the same condition within the same node. This feature saves time when you want to make identical transformations on several fields in your dataset. For example, if you want to build a regression model predicting current salary based on beginning salary and previous experience, it might be beneficial to apply a log transformation to all three skewed variables. Rather than add a new Derive node for each transformation, you can apply the same function to all fields at once. Simply select all fields from which to derive a new field and then type the derive expression using the `@FIELD` function within the field parentheses.

Note: The `@FIELD` function is an important tool for deriving multiple fields at the same time. It allows you to refer to the contents of the current field or fields without specifying the exact field name. For instance, a CLEM expression used to apply a log transformation to multiple fields is `log(@FIELD)`.

The following options are added to the dialog box when you select Multiple mode:

Derive from. Use the Field Chooser to select fields from which to derive new fields. One output field will be generated for each selected field.
Note: Selected fields do not need to be the same storage type; however, the Derive operation will fail if the condition is not valid for *all* fields.

Field name extension. Type the extension that you would like added to the new field name(s). For example, for a new field containing the log of *Current Salary*, you could add the extension *log_* to the field name, producing *log_Current Salary*. Use the radio buttons to choose whether to add the extension as a prefix (at the beginning) or as a suffix (at the end) of the field name. The default name is *DeriveN*, where *N* is the number of Derive nodes that you have created thus far during the current session.

As in the single-mode Derive node, you now need to create an expression to use for deriving a new field. Depending on the type of Derive operation selected, there are a number of options to create a condition. These options are discussed in subsequent topics. To create an expression, you can simply type in the formula field(s) or use the Expression Builder by clicking the calculator button. Remember to use the `@FIELD` function when referring to manipulations on multiple fields.

- [Selecting Multiple Fields](#)

Related information

- [Setting Derive Conditional Options](#)
- [Setting Derive Flag Options](#)
- [Setting Derive Nominal Options](#)
- [Setting Derive State Options](#)
- [Setting Derive Count Options](#)
- [Derive node](#)
- [Setting Basic Options for the Derive Node](#)
- [Recoding Values with the Derive Node](#)

Selecting Multiple Fields

For all nodes that perform operations on multiple input fields, such as Derive (multiple mode), Aggregate, Sort, Multiplot, and Time Plot, you can easily select multiple fields using the Select Fields dialog box.

Sort by. You can sort available fields for viewing by selecting one of the following options:

- **Natural.** View the order of fields as they have been passed down the data stream into the current node.
- **Name.** Use alphabetical order to sort fields for viewing.
- **Type.** View fields sorted by their measurement level. This option is useful when selecting fields with a particular measurement level.

Select fields from the list one at a time or use the Shift-click and Ctrl-click methods to select multiple fields. You can also use the buttons below the list to select groups of fields based on their measurement level, or to select or deselect all fields in the table.

Related information

- [Setting Basic Options for the Derive Node](#)
 - [Deriving Multiple Fields](#)
 - [Setting options for the Aggregate node](#)
 - [Sort Node](#)
 - [Multiplot Plot Tab](#)
 - [Time Plot Tab](#)
-

Setting Derive Formula Options

Derive Formula nodes create a new field for each record in a data set based on the results of a CLEM expression. This expression cannot be conditional. To derive values that are based on a conditional expression, use the flag or conditional type of Derive node.

Formula Specify a formula by using the CLEM language to derive a value for the new field.

Note: Because SPSS® Modeler cannot know what submeasurement level is to be used for a derived list field, for the Collection and Geospatial measurement levels, you can click **Specify...** to open the Values dialog box and set the required submeasurement level. For more information, see [Setting derived list values](#).

For Geospatial fields, the only relevant storage types are Real and Integer (the default setting is Real).

- [Setting derived list values](#)
- [Deriving a list or geospatial field](#)

Related information

- [Setting Basic Options for the Derive Node](#)
 - [Setting derived list values](#)
-

Setting derived list values

The Values dialog box is displayed when you select **Specify...** from the Derive node Formula Field type drop-down list. In this dialog box, you set the submeasurement level values to be used for the Formula Field type measurement levels of either Collection or Geospatial.

Measurement Select either Collection or Geospatial. If you select any other measurement level, the dialog box displays a message that there are no editable values.

Collection

The only item that you can set for the Collection Measurement level is the List measure. By default this measure is set to Typeless but you can select another value to set the measurement level of the elements within the list. You can choose one of the following options:

- Typeless
- Categorical
- Continuous
- Nominal
- Ordinal
- Flag

Geospatial

For the Geospatial Measurement level, you can select the following options to set the measurement level of the elements within the list:

Type Select the measurement sublevel of the geospatial field. The available sublevels are determined by the depth of the list field; the defaults are as follows:

- Point (zero depth)
- LineString (depth of one)
- Polygon (depth of one)
- MultiPoint (depth of one)
- MultiLineString (depth of two)
- MultiPolygon (depth of two)

For more information about sublevels, see [Geospatial measurement sublevels](#).

For more information about list depths, see [List storage and associated measurement levels](#).

Coordinate system This option is only available if you changed the measurement level to geospatial from a non-geospatial level. To apply a coordinate system to your geospatial data, select this check box. By default, the coordinate system set in the Tools > Stream Properties > Options > Geospatial pane is shown. To use a different coordinate system, click the Change button to display the Select Coordinate System dialog box and choose the system that matches your data.

For more information about coordinate systems, see [Setting geospatial options for streams](#).

Deriving a list or geospatial field

There are occasions when data that should be recorded as a list item is imported into SPSS® Modeler with the wrong attributes. For example, as separate geospatial fields, such as an x coordinate and a y coordinate, or a longitude and latitude, as individual rows in a .csv file. In this situation, you must combine the individual fields into a single list field; one way to do this is by using the Derive node.

Note: You must know which is the x (or longitude) field and which is the y (or latitude field) field when you combine geospatial data. You must combine your data so that the resulting list field has the order of elements: [x, y] or [Longitude, Latitude], which are the standard formats for geospatial coordinates.

The following steps show a simple example of deriving a list field.

1. In your stream, attach a Derive node to your Source node.
2. On the Settings tab of the Derive node, select Formula from the Derive as list.
3. In Field Type select either Collection for a non-geospatial list, or Geospatial. By default, SPSS Modeler uses a "best guess" approach to set the correct list detail; you can select Specify... to open the Values dialog box. This dialog can be used for a collection to enter further information about the data in your list, and for geospatial it can be used to set the type of the data and specify the coordinate system of the data.
Note: For geospatial, the coordinate system you specify must exactly match the coordinate system of the data. If this is not the case, geospatial functionality will produce incorrect results.
4. In the Formula pane, enter the formula to combine your data into the correct list format. Alternatively, click the calculator button to open the Expression Builder.
A simple example of a formula to derive a list is [x, y], where x and y are separate fields in the data source. The new derived field that is created is a list where the value for each record is the concatenated x and y values for that record.

Note: Fields that are combined into a list in this way must have the same storage type.

For more information about lists, and depths of lists, see [List storage and associated measurement levels](#).

Setting Derive Flag Options

Derive Flag nodes are used to indicate a specific condition, such as high blood pressure or customer account inactivity. A flag field is created for each record, and when the true condition is met, the flag value for true is added in the field.

True value. Specify a value to include in the flag field for records that match the condition specified below. The default is T.

False value. Specify a value to include in the flag field for records that do *not* match the condition specified below. The default is F.

True when. Specify a CLEM condition to evaluate certain values of each record and give the record a true value or a false value (defined above). Note that the true value will be given to records in the case of non-false numeric values.

Note: To return an empty string, you should type opening and closing quotes with nothing between them, such as "". Empty strings are often used, for example, as the false value in order to enable true values to stand out more clearly in a table. Similarly, quotes should be used if you want a string value that would otherwise be treated as a number

Example

In releases of IBM® SPSS® Modeler prior to 12.0, multiple responses were imported into a single field, with values separated by commas. For example:

```
museum_of_design,institute_of_textiles_and_fashion  
museum_of_design  
archeological_museum  
$null$  
national_art_gallery,national_museum_of_science,other
```

To prepare this data for analysis, you could use the **hassubstring** function to generate a separate flag field for each response with an expression such as:

```
hassubstring(museums,"museum_of_design")
```

Related information

- [Setting Basic Options for the Derive Node](#)
-

Setting Derive Nominal Options

Derive Nominal nodes are used to execute a set of CLEM conditions in order to determine which condition each record satisfies. As a condition is met for each record, a value (indicating which set of conditions was met) will be added to the new, derived field.

Default value. Specify a value to be used in the new field if none of the conditions are met.

Set field to. Specify a value to enter in the new field when a particular condition is met. Each value in the list has an associated condition that you specify in the adjacent column.

If this condition is true. Specify a condition for each member in the set field to list. Use the Expression Builder to select from available functions and fields. You can use the arrow and delete buttons to reorder or remove conditions.

A condition works by testing the values of a particular field in the dataset. As each condition is tested, the values specified above will be assigned to the new field to indicate which, if any, condition was met. If none of the conditions are met, the default value is used.

Related information

- [Setting Basic Options for the Derive Node](#)
-

Setting Derive State Options

Derive State nodes are somewhat similar to Derive Flag nodes. A Flag node sets values depending on the fulfillment of a *single* condition for the current record, but a Derive State node can change the values of a field depending on how it fulfills *two independent* conditions. This means that the value will change (turn on or off) as each condition is met.

Initial state. Select whether to give each record of the new field the On or Off value initially. Note that this value can change as each condition is met.

"On" value. Specify the value for the new field when the On condition is met.

Switch "On" when. Specify a CLEM condition that will change the state to On when the condition is true. Click the calculator button to open the Expression Builder.

"Off" value. Specify the value for the new field when the Off condition is met.

Switch "Off" when. Specify a CLEM condition that will change the state to Off when the condition is false. Click the calculator button to open the Expression Builder.

Note: To specify an empty string, you should type opening and closing quotes with nothing between them, such as "". Similarly, quotes should be used if you want a string value that would otherwise be treated as a number.

Related information

- [Setting Basic Options for the Derive Node](#)
-

Setting Derive Count Options

A Derive Count node is used to apply a series of conditions to the values of a numeric field in the dataset. As each condition is met, the value of the derived count field is increased by a set increment. This type of Derive node is useful for time series data.

Initial value. Sets a value used on execution for the new field. The initial value must be a numeric constant. Use the arrow buttons to increase or decrease the value.

Increment when. Specify the CLEM condition that, when met, will change the derived value based on the number specified in Increment by. Click the calculator button to open the Expression Builder.

Increment by. Set the value used to increment the count. You can use either a numeric constant or the result of a CLEM expression.

Reset when. Specify a condition that, when met, will reset the derived value to the initial value. Click the calculator button to open the Expression Builder.

Related information

- [Setting Basic Options for the Derive Node](#)

Setting Derive Conditional Options

Derive Conditional nodes use a series of If-Then-Else statements to derive the value of the new field.

If. Specify a CLEM condition that will be evaluated for each record upon execution. If the condition is true (or non-false, in the case of numbers), the new field is given the value specified below by the Then expression. Click the calculator button to open the Expression Builder.

Then. Specify a value or CLEM expression for the new field when the If statement above is true (or non-false). Click the calculator button to open the Expression Builder.

Else. Specify a value or CLEM expression for the new field when the If statement above is false. Click the calculator button to open the Expression Builder.

Related information

- [Setting Basic Options for the Derive Node](#)

Recoding Values with the Derive Node

Derive nodes can also be used to recode values, for example by converting a string field with categorical values to a numeric nominal (set) field.

1. For Derive As, select the type of field (Nominal, Flag, etc.) as appropriate.
2. Specify the conditions for recoding values. For example, you could set the value to 1 if `Drug='drugA'`, 2 if `Drug='drugB'`, and so on.

Related information

- [Setting Derive Formula Options](#)
- [Setting Derive Flag Options](#)
- [Setting Derive Nominal Options](#)
- [Setting Derive Count Options](#)
- [Setting Derive State Options](#)
- [Setting Derive Conditional Options](#)
- [Derive node](#)
- [Setting Basic Options for the Derive Node](#)
- [Deriving Multiple Fields](#)

Filler node

Filler nodes are used to replace field values and change storage. You can choose to replace values based on a specified CLEM condition, such as `@BLANK(@FIELD)`. Alternatively, you can choose to replace all blanks or null values with a specific value. Filler nodes are often used in conjunction with the Type node to replace missing values. For example, you can fill blanks with the mean value of a field by specifying an expression such as `@GLOBAL_MEAN`. This expression will fill all blanks with the mean value as calculated by a Set Globals node.

Fill in fields. Using the Field Chooser (button to the right of the text field), select fields from the dataset whose values will be examined and replaced. The default behavior is to replace values depending on the Condition and Replace with expressions specified below. You can also select an alternative method of replacement using the Replace options below.

Note: When selecting multiple fields to replace with a user-defined value, it is important that the field types are similar (all numeric or all symbolic).

Replace. Select to replace the values of the selected field(s) using one of the following methods:

- Based on condition. This option activates the Condition field and Expression Builder for you to create an expression used as a condition for replacement with the value specified.
- Always. Replaces all values of the selected field. For example, you could use this option to convert the storage of income to a string using the following CLEM expression: `(to_string(income))`.
- Blank values. Replaces all user-specified blank values in the selected field. The standard condition `@BLANK(@FIELD)` is used to select blanks. Note: You can define blanks using the Types tab of the source node or with a Type node.
- Null values. Replaces all system null values in the selected field. The standard condition `@NULL(@FIELD)` is used to select nulls.
- Blank and null values. Replaces both blank values and system nulls in the selected field. This option is useful when you are unsure whether or not nulls have been defined as missing values.

Condition. This option is available when you have selected the Based on condition option. Use this text box to specify a CLEM expression for evaluating the selected fields. Click the calculator button to open the Expression Builder.

Replace with. Specify a CLEM expression to give a new value to the selected fields. You can also replace the value with a null value by typing `undef` in the text box. Click the calculator button to open the Expression Builder.

Note: When the field(s) selected are string, you should replace them with a string value. Using the default 0 or another numeric value as the replacement value for string fields will result in an error.

Note that use of the following may change row order:

- Executing in a database via SQL pushback
 - Executing via remote IBM® SPSS® Analytic Server
 - Using functions that run in embedded IBM SPSS Analytic Server
 - Deriving a list (for example, see [Deriving a list or geospatial field](#))
 - Calling any of the spatial functions
- [Storage Conversion Using the Filler Node](#)

Storage Conversion Using the Filler Node

Using the Replace condition of a Filler node, you can easily convert the field storage for single or multiple fields. For example, using the conversion function `to_integer`, you could convert `income` from a string to an integer using the following CLEM expression: `to_integer(income)`.

You can view available conversion functions and automatically create a CLEM expression using the Expression Builder. From the Functions drop-down list, select Conversion to view a list of storage conversion functions. The following conversion functions are available:

- `to_integer(ITEM)`
- `to_real(ITEM)`
- `to_number(ITEM)`
- `to_string(ITEM)`
- `to_time(ITEM)`
- `to_timestamp(ITEM)`
- `to_date(ITEM)`
- `to_datetime(ITEM)`

Converting date and time values. Note that conversion functions (and any other functions that require a specific type of input such as a date or time value) depend on the current formats specified in the stream options dialog box. For example if you want to convert a string field with values *Jan 2003, Feb 2003*, etc. to date storage, select MON YYYY as the default date format for the stream.

Conversion functions are also available from the Derive node, for temporary conversion during a derive calculation. You can also use the Derive node to perform other manipulations such as recoding string fields with categorical values. See the topic [Recoding Values with the Derive Node](#) for more information.

Reclassify Node

The Reclassify node enables the transformation from one set of categorical values to another. Reclassification is useful for collapsing categories or regrouping data for analysis. For example, you could reclassify the values for *Product* into three groups, such as *Kitchenware*, *Bath and Linens*, and *Appliances*. Often, this operation is performed directly from a Distribution node by grouping values and generating a Reclassify node. See the topic [Using a Distribution Node](#) for more information.

Reclassification can be performed for one or more symbolic fields. You can also choose to substitute the new values for the existing field or generate a new field.

Before using a Reclassify node, consider whether another Field Operations node is more appropriate for the task at hand:

- To populate the column values in the Reclassify node, use Type node as a prefix to Reclassify node.
 - To transform numeric ranges into sets using an automatic method, such as ranks or percentiles, you should use a Binning node. See the topic [Binning Node](#) for more information.
 - To classify numeric ranges into sets manually, you should use a Derive node. For example, if you want to collapse salary values into specific salary range categories, you should use a Derive node to define each category manually.
 - To create one or more flag fields based on the values of a categorical field, such as *Mortgage_type*, you should use a Set to Flag node.
 - To convert a categorical field to numeric storage, you can use a Derive node. For example, you could convert No and Yes values to 0 and 1, respectively. See the topic [Recoding Values with the Derive Node](#) for more information.
- [Setting Options for the Reclassify Node](#)
- [Reclassifying Multiple Fields](#)
- [Storage and Measurement Level for Reclassified Fields](#)

Related information

- [Field Operations Overview](#)
- [Setting Options for the Reclassify Node](#)
- [Reclassifying Multiple Fields](#)
- [Storage and Measurement Level for Reclassified Fields](#)

Setting Options for the Reclassify Node

There are three steps to using the Reclassify node:

1. First, select whether you want to reclassify multiple fields or a single field.
2. Next, choose whether to recode into the existing field or create a new field.
3. Then, use the dynamic options in the Reclassify node dialog box to map sets as desired.

Mode. Select Single to reclassify the categories for one field. Select Multiple to activate options enabling the transformation of more than one field at a time.

Reclassify into. Select New field to keep the original nominal field and derive an additional field containing the reclassified values. Select Existing field to overwrite the values in the original field with the new classifications. This is essentially a "fill" operation.

Once you have specified mode and replacement options, you must select the transformation field and specify the new classification values using the dynamic options on the bottom half of the dialog box. These options vary depending on the mode you have selected above.

Reclassify field(s). Use the Field Chooser button on the right to select one (Single mode) or more (Multiple mode) categorical fields.

New field name. Specify a name for the new nominal field containing recoded values. This option is available only in Single mode when New field is selected above. When Existing field is selected, the original field name is retained. When working in Multiple mode, this option is replaced with controls for specifying an extension added to each new field. See the topic [Reclassifying Multiple Fields](#) for more information.

Reclassify values. This table enables a clear mapping from old set values to those you specify here.

- **Original value.** This column lists existing values for the select field(s).
 - **New value.** Use this column to type new category values or select one from the drop-down list. When you automatically generate a Reclassify node using values from a Distribution chart, these values are included in the drop-down list. This allows you to quickly map existing values to a known set of values. For example, healthcare organizations sometimes group diagnoses differently based upon network or locale. After a merger or acquisition, all parties will be required to reclassify new or even existing data in a consistent fashion. Rather than manually typing each target value from a lengthy list, you can read the master list of values in to IBM® SPSS® Modeler, run a Distribution chart for the *Diagnosis* field, and generate a Reclassify (values) node for this field directly from the chart. This process will make all of the target Diagnosis values available from the New Values drop-down list.
4. Click Get to read original values for one or more fields selected above.

5. Click Copy to paste original values over to the *New value* column for fields that have not been mapped yet. The unmapped original values are added to the drop-down list.
6. Click Clear new to erase all specifications in the *New value* column. *Note:* This option does not erase the values from the drop-down list.
7. Click Auto to automatically generate consecutive integers for each of the original values. Only integer values (no real values, such as 1.5, 2.5, and so on) can be generated.

For example, you can automatically generate consecutive product ID numbers for product names or course numbers for university class offerings. This functionality corresponds to the Automatic Recode transformation for sets in IBM SPSS Statistics.

For unspecified values use. This option is used for filling unspecified values in the new field. You can either choose to keep the original value by selecting Original value or specify a default value.

Related information

- [Reclassify Node](#)
 - [Reclassifying Multiple Fields](#)
 - [Storage and Measurement Level for Reclassified Fields](#)
-

Reclassifying Multiple Fields

To map category values for more than one field at a time, set the mode to Multiple. This enables new settings in the Reclassify dialog box, which are described below.

Reclassify fields. Use the Field Chooser button on the right to select the fields that you want to transform. Using the Field Chooser, you can select all fields at once or fields of a similar type, such as nominal or flag.

Field name extension. When recoding multiple fields simultaneously, it is more efficient to specify a common extension added to all new fields rather than individual field names. Specify an extension such as `_recode` and select whether to append or prepend this extension to the original field names.

Related information

- [Reclassify Node](#)
 - [Setting Options for the Reclassify Node](#)
 - [Storage and Measurement Level for Reclassified Fields](#)
-

Storage and Measurement Level for Reclassified Fields

The Reclassify node always creates a nominal field from the recode operation. In some cases, this may change the measurement level of the field when using the Existing field reclassification mode.

The new field's storage (how data are *stored* rather than how they are *used*) is calculated based on the following Settings tab options:

- If unspecified values are set to use a default value, the storage type is determined by examining both the new values as well as the default value and determining the appropriate storage. For example, if all values can be parsed as integers, the field will have the integer storage type.
- If unspecified values are set to use the original values, the storage type is based on the storage of the original field. If all of the values can be parsed as the storage of the original field, then that storage is preserved; otherwise, the storage is determined by finding the most appropriate storage type encompassing both old and new values. For example, reclassifying an integer set { 1, 2, 3, 4, 5 } with the reclassification 4 => 0, 5 => 0 generates a new integer set { 1, 2, 3, 0 }, whereas with the reclassification 4 => "Over 3", 5 => "Over 3" will generate the string set { "1", "2", "3", "Over 3" }.

Note: If the original type was uninstantiated, the new type will be also be uninstantiated.

Related information

- [Reclassify Node](#)
- [Setting Options for the Reclassify Node](#)
- [Reclassifying Multiple Fields](#)
- [Setting Field Storage and Formatting](#)

Anonymize Node

The Anonymize node enables you to disguise field names, field values, or both when working with data that are to be included in a model downstream of the node. In this way, the generated model can be freely distributed (for example, to Technical Support) with no danger that unauthorized users will be able to view confidential data, such as employee records or patients' medical records.

Depending on where you place the Anonymize node in the stream, you may need to make changes to other nodes. For example, if you insert an Anonymize node upstream from a Select node, the selection criteria in the Select node will need to be changed if they are acting on values that have now become anonymized.

The method to be used for anonymizing depends on various factors. For field names and all field values except Continuous measurement levels, the data are replaced by a string of the form:

`prefix_Sn`

where `prefix_` is either a user-specified string or the default string `anon_`, and `n` is an integer value that starts at 0 and is incremented for each unique value (for example, `anon_S0`, `anon_S1`, etc.).

Field values of type Continuous must be transformed because numeric ranges deal with integer or real values rather than strings. As such, they can be anonymized only by transforming the range into a different range, thus disguising the original data. Transformation of a value `x` in the range is performed in the following way:

`A*(x + B)`

where:

`A` is a scale factor, which must be greater than 0.

`B` is a translation offset to be added to the values.

Example

In the case of a field `AGE` where the scale factor `A` is set to 7 and the translation offset `B` is set to 3, the values for `AGE` are transformed into:

`7*(AGE + 3)`

- [Setting Options for the Anonymize Node](#)
- [Anonymizing Field Values](#)

Related information

- [Field Operations Overview](#)
- [Anonymizing Field Names](#)
- [Setting Options for the Anonymize Node](#)
- [Anonymizing Field Values](#)

Setting Options for the Anonymize Node

Here you can choose which fields are to have their values disguised further downstream.

Note that the data fields must be instantiated upstream from the Anonymize node before anonymize operations can be performed. You can instantiate the data by clicking the Read Values button on a Type node or on the Types tab of a source node.

Field. Lists the fields in the current dataset. If any field names have already been anonymized, the anonymized names are shown here.

Measurement. The measurement level of the field.

Anonymize Values. Select one or more fields, click this column, and choose Yes to anonymize the field value using the default prefix `anon_`; choose Specify to display a dialog box in which you can either enter your own prefix or, in the case of field values of type *Continuous*, specify whether the transformation of field values is to use random or user-specified values. Note that *Continuous* and non-*Continuous* field types cannot be specified in the same operation; you must do this separately for each type of field.

View current fields. Select to view fields for datasets actively connected to the Anonymize node. This option is selected by default.

View unused field settings. Select to view fields for datasets that were once but are no longer connected to the node. This option is useful when copying nodes from one stream to another or when saving and reloading nodes.

- [Specifying How Field Values Will Be Anonymized](#)

Specifying How Field Values Will Be Anonymized

The Replace Values dialog box lets you choose whether to use the default prefix for anonymized field values or to use a custom prefix. Clicking OK in this dialog box changes the setting of Anonymize Values on the Settings tab to Yes for the selected field or fields.

Field values prefix. The default prefix for anonymized field values is anon_ ; choose Custom and enter your own prefix if you want a different one.

The Transform Values dialog box is displayed only for fields of type Continuous and allows you to specify whether the transformation of field values is to use random or user-specified values.

Random. Choose this option to use random values for the transformation. Set random seed is selected by default; specify a value in the Seed field, or use the default value.

Fixed. Choose this option to specify your own values for the transformation.

- **Scale by.** The number by which field values will be multiplied in the transformation. Minimum value is 1; maximum is normally 10, but this may be lowered to avoid overflow.
- **Translate by.** The number that will be added to field values in the transformation. Minimum value is 0; maximum is normally 1000, but this may be lowered to avoid overflow.

Related information

- [Anonymize Node](#)

Anonymizing Field Values

Fields that have been selected for anonymization on the Settings tab have their values anonymized:

- When you run the stream containing the Anonymize node
- When you preview the values

To preview the values, click the Anonymize Values button on the Anonymized Values tab. Next, select a field name from the drop-down list.

If the measurement level is Continuous, the display shows the:

- Minimum and maximum values of the original range
- Equation used to transform the values

If the measurement level is anything other than Continuous, the screen displays the original and anonymized values for that field.

If the display appears with a yellow background, this indicates that either the setting for the selected field has changed since the last time the values were anonymized or that changes have been made to the data upstream of the Anonymize node such that the anonymized values may no longer be correct. The current set of values is displayed; click the Anonymize Values button again to generate a new set of values according to the current setting.

Anonymize Values. Creates anonymized values for the selected field and displays them in the table. If you are using random seeding for a field of type Continuous, clicking this button repeatedly creates a different set of values each time.

Clear Values. Clears the original and anonymized values from the table.

Binning Node

The Binning node enables you to automatically create new nominal fields based on the values of one or more existing continuous (numeric range) fields. For example, you can transform a continuous income field into a new categorical field containing income groups of equal width, or as deviations from the mean. Alternatively, you can select a categorical "supervisor" field in order to preserve the strength of the original association between the two fields.

Binning can be useful for a number of reasons, including:

- **Algorithm requirements.** Certain algorithms, such as Naive Bayes, Logistic Regression, require categorical inputs.
- **Performance.** Algorithms such as multinomial logistic may perform better if the number of distinct values of input fields is reduced. For example, use the median or mean value for each bin rather than using the original values.
- **Data Privacy.** Sensitive personal information, such as salaries, may be reported in ranges rather than actual salary figures in order to protect privacy.

A number of binning methods are available. Once you have created bins for the new field, you can generate a Derive node based on the cut points.

When to use a Binning node

Before using a Binning node, consider whether another technique is more appropriate for the task at hand:

- To manually specify cut points for categories, such as specific predefined salary ranges, use a Derive node. See the topic [Derive node](#) for more information.
- To create new categories for existing sets, use a Reclassify node. See the topic [Reclassify Node](#) for more information.

Missing Value Handling

The Binning node handles missing values in the following ways:

- **User-specified blanks.** Missing values specified as blanks are included during the transformation. For example, if you designated –99 to indicate a blank value using the Type node, this value will be included in the binning process. To ignore blanks during binning, you should use a Filler node to replace the blank values with the system null value.
- **System-missing values (\$null\$).** Null values are ignored during the binning transformation and remain nulls after the transformation.

The Settings tab provides options for available techniques. The View tab displays cut points established for data previously run through the node.

- [Setting Options for the Binning Node](#)
- [Fixed-Width Bins](#)
- [Tiles \(Equal Count or Sum\)](#)
- [Rank Cases](#)
- [Mean/Standard Deviation](#)
- [Optimal Binning](#)
- [Previewing the Generated Bins](#)

Related information

- [Field Operations Overview](#)
 - [Setting Options for the Binning Node](#)
-

Setting Options for the Binning Node

Using the Binning node, you can automatically generate bins (categories) using the following techniques:

- Fixed-width binning
- Tiles (equal count or sum)
- Mean and standard deviation
- Ranks
- Optimized relative to a categorical "supervisor" field

The lower part of the dialog box changes dynamically depending on the binning method you select.

Bin fields. Continuous (numeric range) fields pending transformation are displayed here. The Binning node enables you to bin multiple fields simultaneously. Add or remove fields using the buttons on the right.

Binning method. Select the method used to determine cut points for new field bins (categories). Subsequent topics describe the options available in each case.

Bin thresholds. Specify how the bin thresholds are computed.

- **Always recompute.** Cut points and bin allocations are always recomputed when the node is run.
- **Read from Bin Values tab if available.** Cut points and bin allocations are computed only as necessary (for example, when new data has been added).

The following topics discuss options for the available methods of binning.

Related information

- [Fixed-Width Bins](#)
- [Tiles \(Equal Count or Sum\)](#)
- [Rank Cases](#)
- [Mean/Standard Deviation](#)
- [Optimal Binning](#)

- [Previewing the Generated Bins](#)
- [Binning Node](#)

Fixed-Width Bins

When you choose Fixed-width as the binning method, a new set of options is displayed in the dialog box.

Name extension. Specify an extension to use for the generated field(s). *_BIN* is the default extension. You may also specify whether the extension is added to the start (Prefix) or end (Suffix) of the field name. For example, you could generate a new field called *income_BIN*.

Bin width. Specify a value (integer or real) used to calculate the “width” of the bin. For example, you can use the default value, 10, to bin the field Age. Since Age has a range from 18–65, the generated bins would be as shown in the following table.

Table 1. Bins for Age with range 18–65

Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Bin 6
>=13 to <23	>=23 to <33	>=33 to <43	>=43 to <53	>=53 to <63	>=63 to <73

The start of bin intervals is calculated using the lowest scanned value minus half of the bin width (as specified). For example, in the bins shown above, 13 is used to start the intervals according to the following calculation: $18 \text{ [lowest data value]} - 5 \text{ [0.5} \times \text{ (Bin width of 10)]} = 13$.

No. of bins. Use this option to specify an integer used to determine the number of fixed-width bins (categories) for the new field(s).

Once you have executed the Binning node in a stream, you can view the bin thresholds generated by clicking the Preview tab in the Binning node dialog box. See the topic [Previewing the Generated Bins](#) for more information.

Related information

- [Setting Options for the Binning Node](#)
- [Tiles \(Equal Count or Sum\)](#)
- [Rank Cases](#)
- [Mean/Standard Deviation](#)
- [Optimal Binning](#)
- [Previewing the Generated Bins](#)

Tiles (Equal Count or Sum)

The tile binning method creates nominal fields that can be used to split scanned records into percentile groups (or quartiles, deciles, and so on) so that each group contains the same number of records, or the sum of the values in each group is equal. Records are ranked in ascending order based on the value of the specified bin field, so that records with the lowest values for the selected bin variable are assigned a rank of 1, the next set of records are ranked 2, and so on. The threshold values for each bin are generated automatically based on the data and tiling method used.

Tile name extension. Specify an extension used for field(s) generated using standard p-tiles. The default extension is *_TILE* plus *N*, where *N* is the tile number. You may also specify whether the extension is added to the start (Prefix) or end (Suffix) of the field name. For example, you could generate a new field called *income_BIN4*.

Custom tile extension. Specify an extension used for a custom tile range. The default is *_TILEN*. Note that *N* in this case will not be replaced by the custom number.

Available p-tiles are:

- **Quartile.** Generate 4 bins, each containing 25% of the cases.
- **Quintile.** Generate 5 bins, each containing 20% of the cases.
- **Decile.** Generate 10 bins, each containing 10% of the cases.
- **Vingtile.** Generate 20 bins, each containing 5% of the cases.
- **Percentile.** Generate 100 bins, each containing 1% of the cases.
- **Custom N.** Select to specify the number of bins. For example, a value of 3 would produce 3 banded categories (2 cut points), each containing 33.3% of the cases.

Note that if there are fewer discrete values in the data than the number of tiles specified, all tiles will not be used. In such cases, the new distribution is likely to reflect the original distribution of your data.

Tiling method. Specifies the method used to assign records to bins.

- **Record count.** Seeks to assign an equal number of records to each bin.
- **Sum of values.** Seeks to assign records to bins such that the sum of the values in each bin is equal. When targeting sales efforts, for example, this method can be used to assign prospects to decile groups based on value per record, with the highest value prospects in the

top bin. For example, a pharmaceutical company might rank physicians into decile groups based on the number of prescriptions they write. While each decile would contain approximately the same number of scripts, the number of individuals contributing those scripts would not be the same, with the individuals who write the most scripts concentrated in decile 10. Note that this approach assumes that all values are greater than zero, and may yield unexpected results if this is not the case.

Ties. A tie condition results when values on either side of a cut point are identical. For example, if you are assigning deciles and more than 10% of records have the same value for the bin field, then all of them cannot fit into the same bin without forcing the threshold one way or another. Ties can be moved up to the next bin or kept in the current one but must be resolved so that all records with identical values fall into the same bin, even if this causes some bins to have more records than expected. The thresholds of subsequent bins may also be adjusted as a result, causing values to be assigned differently for the same set of numbers based on the method used to resolve ties.

- **Add to next.** Select to move the tie values up to the next bin.
- **Keep in current.** Keeps tie values in the current (lower) bin. This method may result in fewer total bins being created.
- **Assign randomly.** Select to allocate the tie values randomly to a bin. This attempts to keep the number of records in each bin at an equal amount.

Example: Tiling by Record Count

The following table illustrates how simplified field values are ranked as quartiles when tiling by record count. Note the results vary depending on the selected ties option.

Table 1. Tiling by record count example

Values	Add to Next	Keep in Current
10	1	1
13	2	1
15	3	2
15	3	2
20	4	3

The number of items per bin is calculated as:

`total number of value / number of tiles`

In the simplified example above, the desired number of items per bin is 1.25 (5 values / 4 quartiles). The value 13 (being value number 2) straddles the 1.25 desired count threshold and is therefore treated differently depending on the selected ties option. In Add to Next mode, it is added into bin 2. In Keep in Current mode, it is left in bin 1, pushing the range of values for bin 4 outside that of existing data values. As a result, only three bins are created, and the thresholds for each bin are adjusted accordingly, as shown in the following table.

Table 2. Binning example result

Bin	Lower	Upper
1	≥ 10	<15
2	≥ 15	<20
3	≥ 20	≤ 20

Note: The speed of binning by tiles may benefit from enabling parallel processing.

Related information

- [Setting Options for the Binning Node](#)
- [Fixed-Width Bins](#)
- [Rank Cases](#)
- [Mean/Standard Deviation](#)
- [Optimal Binning](#)
- [Previewing the Generated Bins](#)

Rank Cases

When you choose Ranks as the binning method, a new set of options is displayed in the dialog box.

Ranking creates new fields containing ranks, fractional ranks, and percentile values for numeric fields depending on the options specified below.

Rank order. Select Ascending (lowest value is marked 1) or Descending (highest value is marked 1).

Rank. Select to rank cases in ascending or descending order as specified above. The range of values in the new field will be $1-N$, where N is the number of discrete values in the original field. Tied values are given the average of their rank.

Fractional rank. Select to rank cases where the value of the new field equals rank divided by the sum of the weights of the nonmissing cases. Fractional ranks fall in the range of 0–1.

Percentage fractional rank. Each rank is divided by the number of records with valid values and multiplied by 100. Percentage fractional ranks fall in the range of 1–100.

Extension. For all rank options, you can create custom extensions and specify whether the extension is added to the start (Prefix) or end (Suffix) of the field name. For example, you could generate a new field called *income_P_RANK*.

Related information

- [Setting Options for the Binning Node](#)
 - [Fixed-Width Bins](#)
 - [Tiles \(Equal Count or Sum\)](#)
 - [Mean/Standard Deviation](#)
 - [Optimal Binning](#)
 - [Previewing the Generated Bins](#)
-

Mean/Standard Deviation

When you choose Mean/standard deviation as the binning method, a new set of options is displayed in the dialog box.

This method generates one or more new fields with banded categories based on the values of the mean and standard deviation of the distribution of the specified field(s). Select the number of deviations to use below.

Name extension. Specify an extension to use for the generated field(s). *_SDBIN* is the default extension. You may also specify whether the extension is added to the start (Prefix) or end (Suffix) of the field name. For example, you could generate a new field called *income_SDBIN*.

- **+/- 1 standard deviation.** Select to generate three bins.
- **+/- 2 standard deviations.** Select to generate five bins.
- **+/- 3 standard deviations.** Select to generate seven bins.

For example, selecting +/-1 standard deviation results in the three bins as calculated and shown in the following table.

Table 1. Standard deviation bin example

Bin 1	Bin 2	Bin 3
$x < (\text{Mean} - \text{Std. Dev})$	$(\text{Mean} - \text{Std. Dev}) \leq x \leq (\text{Mean} + \text{Std. Dev})$	$x > (\text{Mean} + \text{Std. Dev})$

In a normal distribution, 68% of the cases fall within one standard deviation of the mean, 95% within two standard deviations, and 99% within three standard deviations. Note, however, that creating banded categories based on standard deviations may result in some bins being defined outside the actual data range and even outside the range of possible data values (for example, a negative salary range).

Related information

- [Setting Options for the Binning Node](#)
 - [Fixed-Width Bins](#)
 - [Tiles \(Equal Count or Sum\)](#)
 - [Rank Cases](#)
 - [Optimal Binning](#)
 - [Previewing the Generated Bins](#)
-

Optimal Binning

If the field you want to bin is strongly associated with another categorical field, you can select the categorical field as a "supervisor" field in order to create the bins in such a way as to preserve the strength of the original association between the two fields.

For example, suppose you have used cluster analysis to group states based on delinquency rates for home loans, with the highest rates in the first cluster. In this case, you might choose *Percent past due* and *Percent of foreclosures* as the Bin fields and the cluster membership field generated by the model as the supervisor field.

Name extension Specify an extension to use for the generated field(s) and whether to add it at the start (Prefix) or end (Suffix) of the field name. For example, you could generate a new field called *pastdue_OPTIMAL* and another called *inforeclosure_OPTIMAL*.

Supervisor field A categorical field used to construct the bins.

Pre-bin fields to improve performance with large datasets Indicates if preprocessing should be used to streamline optimal binning. This groups scale values into a large number of bins using a simple unsupervised binning method, represents values within each bin by the mean, and adjusts the case weight accordingly before proceeding with supervised binning. In practical terms, this method trades a degree of precision for speed.

and is recommended for large datasets. You can also specify the maximum number of bins that any variable should end up in after preprocessing when this option is used.

Merge bins that have relatively small case counts with a larger neighbor. If enabled, indicates that a bin is merged if the ratio of its size (number of cases) to that of a neighboring bin is smaller than the specified threshold; note that larger thresholds may result in more merging.

- [Cut Point Settings](#)

Related information

- [Setting Options for the Binning Node](#)
 - [Fixed-Width Bins](#)
 - [Tiles \(Equal Count or Sum\)](#)
 - [Rank Cases](#)
 - [Mean/Standard Deviation](#)
 - [Previewing the Generated Bins](#)
 - [Cut Point Settings](#)
-

Cut Point Settings

The Cut Point Settings dialog box enables you to specify advanced options for the optimal binning algorithm. These options tell the algorithm how to calculate the bins using the target field.

Bin end points. You can specify whether the lower or upper end points should be inclusive (lower $\leq x$) or exclusive (lower $< x$).

First and last bins. For both the first and last bin, you can specify whether the bins should be unbounded (extending toward positive or negative infinity) or bounded by the lowest or highest data points.

Related information

- [Optimal Binning](#)
-

Previewing the Generated Bins

The Bin Values tab in the Binning node allows you to view the thresholds for generated bins. Using the Generate menu, you can also generate a Derive node that can be used to apply these thresholds from one dataset to another.

Binned field. Use the drop-down list to select a field for viewing. Field names shown use the original field name for clarity.

Tile. Use the drop-down list to select a tile, such as 10 or 100, for viewing. This option is available only when bins have been generated using the tile method (equal count or sum).

Bin thresholds. Threshold values are shown here for each generated bin, along with the number of records that fall into each bin. For the optimal binning method only, the number of records in each bin is shown as a percentage of the whole. Note that thresholds are not applicable when the rank binning method is used.

Read Values. Reads binned values from the dataset. Note that thresholds will also be overwritten when new data are run through the stream.

Generating a Derive Node

You can use the Generate menu to create a Derive node based on the current thresholds. This is useful for applying established bin thresholds from one set of data to another. Furthermore, once these split points are known, a Derive operation is more efficient (meaning faster) than a Binning operation when working with large datasets.

Related information

- [Setting Options for the Binning Node](#)
- [Fixed-Width Bins](#)
- [Tiles \(Equal Count or Sum\)](#)
- [Rank Cases](#)
- [Mean/Standard Deviation](#)
- [Optimal Binning](#)

RFM Analysis Node

The Recency, Frequency, Monetary (RFM) Analysis node enables you to determine quantitatively which customers are likely to be the best ones by examining how recently they last purchased from you (recency), how often they purchased (frequency), and how much they spent over all transactions (monetary).

The reasoning behind RFM analysis is that customers who purchase a product or service once are more likely to purchase again. The categorized customer data is separated into a number of bins, with the binning criteria adjusted as you require. In each of the bins, customers are assigned a score; these scores are then combined to provide an overall RFM score. This score is a representation of the customer's membership in the bins created for each of the RFM parameters. This binned data may be sufficient for your needs, for example, by identifying the most frequent, high-value customers; alternatively, it can be passed on in a stream for further modeling and analysis.

Note, however, that although the ability to analyze and rank RFM scores is a useful tool, you must be aware of certain factors when using it. There may be a temptation to target customers with the highest rankings; however, over-solicitation of these customers could lead to resentment and an actual fall in repeat business. It is also worth remembering that customers with low scores should not be neglected but instead may be cultivated to become better customers. Conversely, high scores alone do not necessarily reflect a good sales prospect, depending on the market. For example, a customer in bin 5 for recency, meaning that they have purchased very recently, may not actually be the best target customer for someone selling expensive, longer-life products such as cars or televisions.

Note: Depending on how your data are stored, you may need to precede the RFM Analysis node with an RFM Aggregate node to transform the data into a usable format. For example, input data must be in customer format, with one row per customer; if the customers' data are in transactional form, use an RFM Aggregate node upstream to derive the recency, frequency, and monetary fields. See the topic [RFM Aggregate Node](#) for more information.

The RFM Aggregate and RFM Analysis nodes in IBM® SPSS® Modeler are set up to use independent binning; that is, they rank and bin data on each measure of recency, frequency, and monetary value, without regard to their values or the other two measures.

- [RFM Analysis Node Settings](#)
- [RFM Analysis Node Binning](#)

Related information

- [Field Operations Overview](#)
- [RFM Analysis Node Settings](#)
- [RFM Analysis Node Binning](#)

RFM Analysis Node Settings

Recency. Using the Field Chooser (button to the right of the text box), select the recency field. This may be a date, timestamp, or simple number. Note that when a date or timestamp represents the date of the most recent transaction, the highest value is considered the most recent; where a number is specified, it represents the time elapsed since the most recent transaction and the lowest value is considered as the most recent.

Note: If the RFM Analysis node is preceded in the stream by an RFM Aggregate node, the Recency, Frequency, and Monetary fields generated by the RFM Aggregate node should be selected as inputs in the RFM Analysis node.

Frequency. Using the Field Chooser, select the frequency field to be used.

Monetary. Using the Field Chooser, select the monetary field to be used.

Number of bins. For each of the three output types, select the number of bins to be created. The default is 5.

Note: The minimum number of bins is 2, and the maximum is 9.

Weight. By default, the highest importance when calculating scores is given to the recency data, followed by frequency, and then monetary. If required, you can amend the weighting affecting one or several of these to change which is given the highest importance.

The RFM score is calculated as follows: (Recency score x Recency weight) + (Frequency score x Frequency weight) + (Monetary score x Monetary weight).

Ties. Specify how identical (tied) scores are to be binned. The options are:

- **Add to next.** Select to move the tie values up to the next bin.
- **Keep in current.** Keeps tie values in the current (lower) bin. This method may result in fewer total bins being created. (This is the default value.)

Bin thresholds. Specify whether the RFM scores and bin allocations are always recomputed when the node is executed or that they are computed only as necessary (for example, when new data has been added). If you select Read from Bin Values tab if available you can edit the

upper and lower cut points for the different bins on the Bin Values tab.

When executed, the RFM Analysis node bins the raw recency, frequency, and monetary fields and adds the following new fields to the dataset:

- Recency score. A rank (bin value) for Recency
- Frequency score. A rank (bin value) for Frequency
- Monetary score. A rank (bin value) for Monetary
- RFM score. The weighted sum of the recency, frequency, and monetary scores.

Add outliers to end bins. If you select this check box, records that lie below the lower bin are added to the lower bin, and records above the highest bin are added to the highest bin—otherwise, they are given a null value. This box is available only if you select Read from Bin Values tab if available.

Related information

- [RFM Analysis Node](#)
 - [RFM Analysis Node Binning](#)
-

RFM Analysis Node Binning

The Bin Values tab allows you to view, and in certain cases amend, the thresholds for generated bins.

Note: You can only amend values on this tab if you select Read from Bin Values tab if available on the Settings tab.

Binned field. Use the drop-down list to select a field for dividing into bins. The available values are those selected on the Settings tab.

Bin values table. Threshold values are shown here for each generated bin. If you select Read from Bin Values tab if available on the Settings tab, you can amend the upper and lower cut points for each bin by double-clicking on the relevant cell.

Read Values. Reads binned values from the dataset and populates the bin values table. Note that if you select Always recompute on the Settings tab, the bin thresholds will be overwritten when new data are run through the stream.

Related information

- [RFM Analysis Node](#)
 - [RFM Analysis Node Settings](#)
-

Ensemble Node

The Ensemble node combines two or more model nuggets to obtain more accurate predictions than can be gained from any of the individual models. By combining predictions from multiple models, limitations in individual models may be avoided, resulting in a higher overall accuracy. Models combined in this manner typically perform at least as well as the best of the individual models and often better.

This combining of nodes happens automatically in the Auto Classifier, Auto Numeric and Auto Cluster automated modeling nodes.

After using an Ensemble node, you can use an Analysis node or Evaluation node to compare the accuracy of the combined results with each of the input models. To do this, make sure the Filter out fields generated by ensembled models option is not selected on the Settings tab in the Ensemble node.

Output Fields

Each Ensemble node generates a field containing the combined scores. The name is based on the specified target field and prefixed with \$XF_, \$XS_, or \$XR_, depending on the field measurement level--flag, nominal (set), or continuous (range), respectively. For example, if the target is a flag field named *response*, the output field would be \$XF_*response*.

Confidence or propensity fields. For flag and nominal fields, additional confidence or propensity fields are created based on the ensemble method, as detailed in the following table.

Table 1. Ensemble method field creation

Ensemble method	Field name
-----------------	------------

Ensemble method	Field name
Voting	<code>\$XFC_<field></code>
Confidence-weighted voting	
Raw-propensity-weighted voting	
Adjusted-propensity-weighted voting	
Highest confidence wins	
Average raw propensity	<code>\$XFNP_<field></code>
Average adjusted raw propensity	<code>\$XFAP_<field></code>

- [Ensemble Node Settings](#)
-

Ensemble Node Settings

Target field for ensemble. Select a single field that is used as the target by two or more upstream models. The upstream models can use flag, nominal, or continuous targets, but at least two of the models must share the same target in order to combine scores.

Filter out fields generated by ensembled models. Removes from the output all of the additional fields generated by the individual models that feed into the Ensemble node. Select this check box if you are interested only in the combined score from all of the input models. Ensure that this option is deselected if, for example, you want to use an Analysis node or Evaluation node to compare the accuracy of the combined score with that of each of the individual input models.

Available settings depend on the measurement level of the field selected as the target.

Continuous Targets

For a continuous target, scores will be averaged. This is the only available method for combining scores.

When averaging scores or estimates, the Ensemble node uses a standard error calculation to work out the difference between the measured or estimated values and the true values, and to show how close those estimates matched. Standard error calculations are generated by default for new models; however, you can deselect the check box for existing models, for example, if they are to be regenerated.

Categorical Targets

For categorical targets, a number of methods are supported, including **voting**, which works by tallying the number of times each possible predicted value is chosen and selecting the value with the highest total. For example, if three out of five models predict yes and the other two predict no, then yes wins by a vote of 3 to 2. Alternatively, votes can be **weighted** based on the confidence or propensity value for each prediction. The weights are then summed, and the value with the highest total is again selected. The confidence for the final prediction is the sum of the weights for the winning value divided by the number of models included in the ensemble.

All categorical fields. For both flag and nominal fields, the following methods are supported:

- Voting
- Confidence-weighted voting
- Highest confidence wins

Flag fields only. For flag fields only, a number of methods based on propensity are also available:

- Raw propensity-weighted voting
- Adjusted propensity-weighted voting
- Average raw propensity
- Average adjusted propensity

Voting ties. For voting methods, you can specify how ties are resolved.

- **Random selection.** One of the tied values is chosen at random.
- **Highest confidence.** The tied value that was predicted with the highest confidence wins. Note that this is not necessarily the same as the highest confidence of all predicted values.
- **Raw or adjusted propensity (flag fields only).** The tied value that was predicted with the largest absolute propensity, where the absolute propensity is calculated as:

```
abs(0.5 - propensity) *
2
```

Or, in the case of adjusted propensity:

```
abs(0.5 - adjusted propensity) * 2
```

Partition Node

Partition nodes are used to generate a partition field that splits the data into separate subsets or samples for the training, testing, and validation stages of model building. By using one sample to generate the model and a separate sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data.

The Partition node generates a nominal field with the role set to Partition. Alternatively, if an appropriate field already exists in your data, it can be designated as a partition using a Type node. In this case, no separate Partition node is required. Any instantiated nominal field with two or three values can be used as a partition, but flag fields cannot be used. See the topic [Setting the field role](#) for more information.

Multiple partition fields can be defined in a stream, but if so, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.)

Enabling partitioning. To use the partition in an analysis, partitioning must be enabled on the Model Options tab in the appropriate model-building or analysis node. Deselecting this option makes it possible to disable partitioning without removing the field.

To create a partition field based on some other criterion such as a date range or location, you can also use a Derive node. See the topic [Derive node](#) for more information.

Example. When building an RFM stream to identify recent customers who have positively responded to previous marketing campaigns, the marketing department of a sales company uses a Partition node to split the data into training and test partitions.

- [Partition Node Options](#)

Related information

- [Field Operations Overview](#)
 - [Partition Node Options](#)
-

Partition Node Options

Partition field. Specifies the name of the field created by the node.

Partitions. You can partition the data into two samples (train and test) or three (train, test, and validation).

- Train and test. Partitions the data into two samples, allowing you to train the model with one sample and test with another.
- Train, test, and validation. Partitions the data into three samples, allowing you to train the model with one sample, test and refine the model using a second sample, and validate your results with a third. This reduces the size of each partition accordingly, however, and may be most suitable when working with a very large dataset.

Partition size. Specifies the relative size of each partition. If the sum of the partition sizes is less than 100%, then the records not included in a partition will be discarded. For example, if a user has 10 million records and has specified partition sizes of 5% training and 10% testing, after running the node, there should be roughly 500,000 training and one million testing records, with the remainder having been discarded.

Values. Specifies the values used to represent each partition sample in the data.

- Use system-defined values ("1," "2," and "3"). Uses an integer to represent each partition; for example, all records that fall into the training sample have a value of 1 for the partition field. This ensures the data will be portable between locales and that if the partition field is reinstated elsewhere (for example, reading the data back from a database), the sort order is preserved (so that 1 will still represent the training partition). However, the values do require some interpretation.
- Append labels to system-defined values. Combines the integer with a label; for example, training partition records have a value of **1_Training**. This makes it possible for someone looking at the data to identify which value is which, and it preserves sort order. However, values are specific to a given locale.
- Use labels as values. Uses the label with no integer; for example, Training. This allows you to specify the values by editing the labels. However, it makes the data locale-specific, and reinstatement of a partition column will put the values in their natural sort order, which may not correspond to their "semantic" order.

Seed. Only available when Repeatable partition assignment is selected. When sampling or partitioning records based on a random percentage, this option allows you to duplicate the same results in another session. By specifying the starting value used by the random number generator, you can ensure the same records are assigned each time the node is executed. Enter the desired seed value, or click the Generate button to automatically generate a random value. If this option is not selected, a different sample will be generated each time the node is executed.

Note: When using the Seed option with records read from a database, a Sort node may be required prior to sampling in order to ensure the same result each time the node is executed. This is because the random seed depends on the order of records, which is not guaranteed to stay the same in a relational database. See the topic [Sort Node](#) for more information.

Use unique field to assign partitions. Only available when Repeatable partition assignment is selected. (For Tier 1 databases only) Check this box to use SQL pushback to assign records to partitions. From the drop-down list, choose a field with unique values (such as an ID field) to ensure that records are assigned in a random but repeatable way.

Database tiers are explained in the description of the Database source node. See the topic [Database source node](#) for more information.

Generating select nodes

Using the Generate menu in the Partition node, you can automatically generate a Select node for each partition. For example, you could select all records in the training partition to obtain further evaluation or analyses using only this partition.

Related information

- [Partition Node](#)
-

Set to Flag Node

The Set to Flag node is used to derive flag fields based on the categorical values defined for one or more nominal fields. For example, your dataset might contain a nominal field, *BP* (blood pressure), with the values *High*, *Normal*, and *Low*. For easier data manipulation, you might create a flag field for high blood pressure, which indicates whether or not the patient has high blood pressure.

- [Setting Options for the Set to Flag Node](#)
-

Related information

- [Field Operations Overview](#)
 - [Setting Options for the Set to Flag Node](#)
-

Setting Options for the Set to Flag Node

Set fields. Lists all data fields with a measurement level of *Nominal* (set). Select one from the list to display the values in the set. You can choose from these values to create a flag field. Note that data must be fully instantiated using an upstream source or Type node before you can see the available nominal fields (and their values). See the topic [Type Node](#) for more information.

Field name extension. Select to enable controls for specifying an extension that will be added as a suffix or prefix to the new flag field. By default, new field names are automatically created by combining the original field name with the field value into a label, such as *Fieldname_fieldvalue*.

Available set values. Values in the set selected above are displayed here. Select one or more values for which you want to generate flags. For example, if the values in a field called *blood_pressure* are *High*, *Medium*, and *Low*, you can select *High* and add it to the list on the right. This will create a field with a flag for records with a value indicating high blood pressure.

Create flag fields. The newly created flag fields are listed here. You can specify options for naming the new field using the field name extension controls.

True value. Specify the true value used by the node when setting a flag. By default, this value is T.

False value. Specify the false value used by the node when setting a flag. By default, this value is F.

Aggregate keys. Select to group records together based on key fields specified below. When Aggregate keys is selected, all flag fields in a group will be "turned on" if *any* record was set to true. Use the Field Chooser to specify which key fields will be used to aggregate records.

Related information

- [Set to Flag Node](#)
-

Restructure Node

The Restructure node can be used to generate multiple fields based on the values of a nominal or flag field. The newly generated fields can contain values from another field or numeric flags (0 and 1). The functionality of this node is similar to that of the Set to Flag node. However, it offers more flexibility. It allows you to create fields of any type (including numeric flags), using the values from another field. You can then perform aggregation or other manipulations with other nodes downstream. (The Set to Flag node lets you aggregate fields in one step, which may be convenient if you are creating flag fields.)

For example, the following dataset contains a nominal field, *Account*, with the values *Savings* and *Draft*. The opening balance and current balance are recorded for each account, and some customers have multiple accounts of each type. Let's say you want to know whether each customer has

a particular account type, and if so, how much money is in each account type. You use the Restructure node to generate a field for each of the *Account* values, and you select *Current_Balance* as the value. Each new field is populated with the current balance for the given record.

Table 1. Sample data before restructuring

CustID	Account	Open_Bal	Current_Bal
12701	Draft	1000	1005.32
12702	Savings	100	144.51
12703	Savings	300	321.20
12703	Savings	150	204.51
12703	Draft	1200	586.32

Table 2. Sample data after restructuring

CustID	Account	Open_Bal	Current_Bal	Account_Draft_Current_Bal	Account_Savings_Current_Bal
12701	Draft	1000	1005.32	1005.32	\$null\$
12702	Savings	100	144.51	\$null\$	144.51
12703	Savings	300	321.20	\$null\$	321.20
12703	Savings	150	204.51	\$null\$	204.51
12703	Draft	1200	586.32	586.32	\$null\$

Using the Restructure Node with the Aggregate Node

In many cases, you may want to pair the Restructure node with an Aggregate node. In the previous example, one customer (with the ID 12703) has three accounts. You can use an Aggregate node to calculate the total balance for each account type. The key field is *CustID*, and the aggregate fields are the new restructured fields, *Account_Draft_Current_Bal* and *Account_Savings_Current_Bal*. The following table shows the results.

Table 3. Sample data after restructuring and aggregation

CustID	Record_Count	Account_Draft_Current_Bal_Sum	Account_Savings_Current_Bal_Sum
12701	1	1005.32	\$null\$
12702	1	\$null\$	144.51
12703	3	586.32	525.71

- [Setting Options for the Restructure Node](#)

Related information

- [Field Operations Overview](#)
- [Setting Options for the Restructure Node](#)

Setting Options for the Restructure Node

Available fields. Lists all data fields with a measurement level of *Nominal* (set) or *Flag*. Select one from the list to display the values in the set or flag, then choose from these values to create the restructured fields. Note that data must be fully instantiated using an upstream source or Type node before you can see the available fields (and their values). See the topic [Type Node](#) for more information.

Available values. Values in the set selected above are displayed here. Select one or more values for which you want to generate restructured fields. For example, if the values in a field called *Blood Pressure* are *High*, *Medium*, and *Low*, you can select *High* and add it to the list on the right. This will create a field with a specified value (see below) for records with a value of *High*.

Create restructured fields. The newly created restructured fields are listed here. By default, new field names are automatically created by combining the original field name with the field value into a label, such as *Fieldname_fieldvalue*.

Include field name. Deselect to remove the original field name as a prefix from the new field names.

Use values from other fields. Specify one or more fields whose values will be used to populate the restructured fields. Use the Field Chooser to select one or more fields. For each field chosen, one new field is created. The value field name is appended to the restructured field name—for example, *BP_High_Age* or *BP_Low_Age*. Each new field inherits the type of the original value field.

Create numeric value flags. Select to populate the new fields with numeric value flags (0 for false and 1 for true), rather than using a value from another field.

Related information

- [Restructure Node](#)
-

Transpose Node

By default, columns are fields and rows are records or observations. If necessary, you can use a Transpose node to swap the data in rows and columns so that fields become records and records become fields. For example, if you have time series data where each series is a row rather than a column, you can transpose the data prior to analysis.

- [Setting options for the Transpose node](#)
-

Setting options for the Transpose node

In the Transpose method drop-down, select the method you want the Transpose node to perform: Both fields and records, Records to fields, or Fields to records. The settings for each of the three methods are described in the following sections.

Restriction: The Records to fields and Fields to records methods are only supported on Windows 64-bit, Linux 64-bit, and Mac.

Both fields and records

New field names can be generated automatically based on a specified prefix, or read from an existing field in the data.

Use prefix. This option generates new field names automatically based on the specified prefix (**Field1**, **Field2**, and so on). You can customize the prefix as needed. With this option, you must specify the number of fields to be created, regardless of the number of rows in the original data. For example, if Number of new fields is set to 100, all data beyond the first 100 rows will be discarded. If there are fewer than 100 rows in the original data, some fields will be null. (You can increase the number of fields as needed, but the purpose of this setting is to avoid transposing a million records into a million fields, which would produce an unmanageable result.)

For example, suppose you have data with series in rows and a separate field (column) for each month. You can transpose this so that each series is in a separate field, with a row for each month.

Read from field. Reads field names from an existing field. With this option, the number of new fields is determined by the data, up to the specified maximum. Each value of the selected field becomes a new field in the output data. The selected field can have any storage type (integer, string, date, and so on), but in order to avoid duplicate field names, each value of the selected field must be unique (in other words, the number of values should match the number of rows). If duplicate field names are encountered, a warning is displayed.

- Read Values. If the selected field has not been instantiated, select this option to populate the list of new field names. If the field has already been instantiated, then this step is not necessary.
- Maximum number of values to read. When reading fields names from the data, an upper limit is specified in order to avoid creating an inordinately large number of fields. (As noted above, transposing one million records into one million fields would produce an unmanageable result.)

For example, if the first column in your data specifies the name for each series, you can use these values as fields names in the transposed data.

Transpose. By default, only continuous (numeric range) fields are transposed (either integer or real storage). Optionally, you can choose a subset of numeric fields or transpose string fields instead. However, all transposed fields must be of the same storage type—either numeric or string but not both—since mixing the input fields would generate mixed values within each output column, which violates the rule that all values of a field must have the same storage. Other storage types (date, time, timestamp) cannot be transposed.

- All numeric. Transposes all numeric fields (integer or real storage). The number of rows in the output matches the number of numeric fields in the original data.
- All string. Transposes all string fields.
- Custom. Allows you to select a subset of numeric fields. The number of rows in the output matches the number of fields selected. This option is only available for numeric fields.

Row ID name. Specifies the name of the row ID field created by the node. The values of this field are determined by the names of the fields in the original data.

Tip: When transposing time series data from rows to columns, if your original data includes a row, such as date, month, or year, that labels the period for each measurement, be sure to read these labels into IBM® SPSS® Modeler as field names (as demonstrated in the above examples, which show the month or date as field names in the original data, respectively) rather than including the label in the first row of data. This will avoid mixing labels and values in each column (which would force numbers to be read as strings, since storage types cannot be mixed within a column).

Records to fields

Fields. The Fields list contains all fields entering the Transpose node.

Index. Use the Index section to select the fields you want to use as index fields.

Fields. Use the Fields section to select the fields you want to use as fields.

Value. Use the Value section to select the fields you want to use as value fields.

Aggregate function. When there are more than one records for an index, you must aggregate the records into one. Use the Aggregate function drop-down to specify how to aggregate the records using one of the following functions. Note that aggregation impacts all fields.

- **Mean.** Returns the mean values for each key field combination. The mean is a measure of central tendency, and is the arithmetic average (the sum divided by the number of cases).
- **Sum.** Returns summed values for each key field combination. The sum is the total of the values, across all cases with nonmissing values.
- **Min.** Returns minimum values for each key field combination.
- **Max.** Returns maximum values for each key field combination.
- **Median.** Returns the median values for each key field combination. The median is a measure of central tendency that is not sensitive to outlying values (unlike the mean, which can be affected by a few extremely high or low values). Also known as the 50th percentile or 2nd quartile.
- **Count.** Returns the count of non-null values for each key field combination.

Fields to records

Fields. The Fields list contains all fields entering the Transpose node.

Index. Use the Index section to select the fields you want to use as index fields.

Value. Use the Value section to select the fields you want to use as value fields. If you don't select any value fields, then all unassigned numeric fields will be used as values. But if not numeric fields are available, then all unassigned string fields will be used.

History Node

History nodes are most often used for sequential data, such as time series data. They are used to create new fields containing data from fields in previous records. When using a History node, you may want to have data that is presorted by a particular field. You can use a Sort node to do this.

- [Setting Options for the History Node](#)

Related information

- [Field Operations Overview](#)
- [Setting Options for the History Node](#)

Setting Options for the History Node

Selected fields. Using the Field Chooser (button to the right of the text box), select the fields for which you want a history. Each selected field is used to create new fields for all records in the dataset.

Offset. Specify the latest record prior to the current record from which you want to extract historical field values. For example, if Offset is set to 3, as each record passes through this node, the field values for the third record previous will be included in the current record. Use the Span settings to specify how far back records will be extracted from. Use the arrows to adjust the offset value.

Span. Specify how many prior records from which you want to extract values. For example, if Offset is set to 3 and Span is set to 5, each record that passes through the node will have five fields added to it for each field specified in the Selected Fields list. This means that when the node is processing record 10, fields will be added from record 7 through record 3. Use the arrows to adjust the span value.

Where history is unavailable. Select one of the following options for handling records that have no history values. This usually refers to the first several records at the top of the dataset, for which there are no previous records to use as a history.

- **Discard records.** Select to discard records where no history value is available for the field selected.
- **Leave history undefined.** Select to keep records where no history value is available. The history field will be filled with an undefined value, displayed as `$null$`.
- **Fill values with.** Specify a value or string to be used for records where no history value is available. The default replacement value is `undef`, the system null. Null values are displayed using the string `$null$`.

When selecting a replacement value, keep in mind the following rules in order for proper execution to occur:

- Selected fields should be of the same storage type.

- If all of the selected fields have numeric storage, the replacement value must be parsed as an integer.
- If all of the selected fields have real storage, the replacement value must be parsed as a real.
- If all of the selected fields have symbolic storage, the replacement value must be parsed as a string.
- If all of the selected fields have date/time storage, the replacement value must be parsed as a date/time field.

If any of the above conditions are not met, you will receive an error when executing the History node.

Related information

- [History Node](#)

Field Reorder Node

The Field Reorder node enables you to define the natural order used to display fields downstream. This order affects the display of fields in a variety of places, such as tables, lists, and the Field Chooser. This operation is useful, for example, when working with wide datasets to make fields of interest more visible.

- [Setting Field Reorder Options](#)

Related information

- [Field Operations Overview](#)
- [Setting Field Reorder Options](#)

Setting Field Reorder Options

There are two ways to reorder fields: custom ordering and automatic sorting.

Custom Ordering

Select Custom Order to enable a table of field names and types where you can view all fields and use arrow buttons to create a custom order.

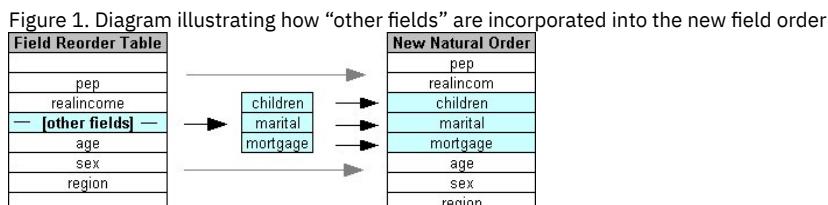
To reorder fields:

1. Select a field in the table. Use the Ctrl-click method to select multiple fields.
2. Use the simple arrow buttons to move the field(s) up or down one row.
3. Use the line-arrow buttons to move the field(s) to the bottom or top of the list.
4. Specify the order of fields not included here by moving up or down the divider row, indicated as [other fields].

More information on [other fields]

Other fields. The purpose of the [other fields] divider row is to break the table into two halves.

- Fields appearing above the divider row will be ordered (as they appear in the table) at the top of all natural orders used to display the fields downstream of this node.
- Fields appearing below the divider row will be ordered (as they appear in the table) at the bottom of all natural orders used to display the fields downstream of this node.



- All other fields not appearing in the field reorder table will appear between these “top” and “bottom” fields as indicated by the placement of the divider row.

Additional custom sorting options include:

- Sort fields in ascending or descending order by clicking on the arrows above each column header (Type, Name, and Storage). When sorting by column, fields not specified here (indicated by the [other fields] row) are sorted last in their natural order.
- Click Clear Unused to delete all unused fields from the Field Reorder node. Unused fields are displayed in the table with a red font. This indicates that the field has been deleted in upstream operations.

- Specify ordering for any new fields (displayed with a lightning icon to indicate a new or unspecified field). When you click OK or Apply, the icon disappears.

Note: If fields are added upstream after a custom order has been applied, the new fields will be appended at the bottom of the custom list.

Automatic Sorting

Select Automatic Sort to specify a parameter for sorting. The dialog box options dynamically change to provide options for automatic sorting.

Sort By. Select one of three ways to sort fields read into the Reorder node. The arrow buttons indicate whether the order will be ascending or descending. Select one to make a change.

- Name
- Type
- Storage

Fields added upstream of the Field Reorder node after auto-sort has been applied will automatically be placed in their proper position based on the sort type selected.

Related information

- [Field Reorder Node](#)
-

Time Intervals Node

The original Time Intervals node, that was available in SPSS® Modeler version 17.1 and earlier, was not compatible with Analytic Server (AS) and was deprecated in SPSS Modeler release 18.0.

The replacement Time Intervals node contains a number of changes from the original Time Intervals node. This new node can be used either with Analytic Server or in SPSS Modeler by itself.

You use the Time Intervals node to specify intervals and derive a new time field for estimating or forecasting. A full range of time intervals is supported, from seconds to years.

Use the node to derive a new time field; the new field has the same storage type as the input time field you chose. The node generates the following items:

- The field specified on the Fields tab as the Time Field, along with the chosen prefix/suffix. By default the prefix is \$TI_.
- The fields specified on the Fields tab as the Dimension fields.
- The fields specified on the Fields tab as the Fields to aggregate.

A number of extra fields can also be generated, depending on the selected interval or period (such as the minute or second within which a measurement falls).

- [Time Interval - field options](#)
 - [Time Interval - build options](#)
-

Time Interval - field options

Use the Fields tab in the Time Intervals node to select the data from which the new time interval is derived.

Fields Displays all of the input fields to the node with their measurement type icons. All time fields have the ‘continuous’ measurement type. Select the field to be used as the input.

Time Field Shows the input field from which the new time interval is derived; only a single continuous field is allowed. The field is used by the Time Intervals node as the aggregation key for converting the interval. The new field has the same storage type as the input time field chosen. If you select an integer field, it is considered to be a time index.

Dimensions fields Optionally, you can add fields here to create an individual time series that is based on the field values. As a simple example, with geospatial data, you can use a point field as a dimension. In this example, the data output from the Time Intervals node is sorted into time series for each point value in the point field.

Dimensions are ideal when you use flattened multi-dimensional data, similar to that generated by the TM1 node, or for supporting more complex data types like geospatial. Essentially, you can consider the use of the Dimensions fields as the equivalent of a **Group By** clause in a SQL query , or similar to Key fields in the Aggregate node; however, the Dimensions fields is more sophisticated in nature due to its ability to handle more complicated data structures than just traditional row and column data.

Fields to aggregate Select the fields to be aggregated as part of changing the period of the time field. Only the fields that you select here are available on the Build tab for the Custom settings for specified fields table. Any fields not included are filtered out of the data that leaves the node. This means that any fields remaining in the Fields list are filtered out of the data.

Related information

- [Overview of Record Operations](#)
 - [Time Intervals Node](#)
 - [Time Interval - build options](#)
-

Time Interval - build options

Use the Build tab to specify options for changing the time interval and how fields in the data are aggregated, based on their measurement type.

When you aggregate data, any existing date, time, or timestamp fields are superseded by the generated fields and are dropped from the output. Other fields are aggregated based on the options you specify in this tab.

Time Interval Select the interval and periodicity for building the series.

For more information, see [Supported Intervals](#).

Default settings Select the default aggregation to be applied to data of different types. The default is applied based on the measurement level; for example, continuous fields are aggregated by the sum, while nominal fields use the mode. You can set the default for 3 different measurement levels:

- **Continuous** Available functions for continuous fields include Sum, Mean, Min, Max, Median, 1st Quartile, and 3rd Quartile.
- **Nominal** Options include Mode, Min, and Max.
- **Flag** Options are either True if any true or False if any false.

Custom settings for specified fields You can specify exceptions to the default aggregation settings for individual fields. Use the icons on the right to add or remove fields from the table, or click the cell in the appropriate column to change the aggregation function for that field. Typeless fields are excluded from the list and cannot be added to the table.

New field name extension Specify the Prefix or Suffix applied to all fields generated by the node.

Related information

- [Overview of Record Operations](#)
 - [Time Intervals Node](#)
 - [Time Interval - field options](#)
-

Reprojection Node

With geospatial or map data, two of the most common ways that are used to identify coordinates are projected coordinate and geographic coordinate systems. Within IBM® SPSS® Modeler, items such as the Expression Builder spatial functions, the Spatio-Temporal Prediction (STP) Node, and the Map Visualization Node use the projected coordinate system and therefore any data that you import that is recorded with a geographic coordinate system must be reprojected. Where possible, geospatial fields (any fields with a geospatial measurement level) are automatically reprojected when used (not when they are imported). Where any fields cannot be reprojected automatically you use the Reproject node to change their coordinate system. Reprojecting in this way means that you can correct situations where an error occurs due to use of an incorrect coordinate system.

Example situations where you might have to reproject to change the coordinate system are shown in the following list:

- **Append** If you try to append two data sets with different coordinate systems for a geospatial field SPSS Modeler displays the following error message: Coordinate systems of <Field1> and <Field2> are not compatible. Reproject one or both fields to the same coordinate system.

<Field1> and <Field2> are the names of the geospatial fields that caused the error.
- **If/else expression** If you use an expression that contains an if/else statement with geospatial fields or return types in both parts of the expression, but with different coordinate systems, SPSS Modeler displays the following error message: The conditional expression contains incompatible coordinate systems: <arg1> and <arg2>.

<arg1> and <arg2> are the then or else arguments that return a geospatial type with different coordinate systems.

- *Constructing a list of geospatial fields* To create a list field that consists of numerous geospatial fields, all geospatial field arguments that are supplied to the list expression must be in the same coordinate system. If they are not, then the following error message is displayed: Coordinate systems of <Field1> and <Field2> are not compatible. Reproject one or both fields to the same coordinate system.

For more information about coordinate systems, see [Setting geospatial options for streams](#).

- [Setting options for the Reproject node](#)

Setting options for the Reproject node

Fields

Geo fields

By default this list is empty. You can move geospatial fields into this list from the Fields to be reprojected list to ensure that those fields are not reprojected.

Fields to be reprojected

By default this list contains all geospatial fields that are input to this node. All fields in this list are reprojected to the coordinate system that you set in the Coordinate System area.

Coordinate System

Stream default

Select this option to use the default coordinate system.

Specify

If you select this option you can use the Change button to display the Select Coordinate System dialog box and choose the coordinate system to be used for reprojection.

For more information about coordinate systems, see [Setting geospatial options for streams](#).

Related information

- [Reprojection Node](#)

Graph Nodes

- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Converting and Distributing Map Shapefiles](#)
- [Plot Node](#)
- [Multiplot Node](#)
- [Time Plot Node](#)
- [Distribution Node](#)
- [Histogram Node](#)
- [Collection Node](#)
- [Web Node](#)
- [Evaluation node](#)
- [Map Visualization Node](#)
- [t-SNE node](#)
- [E-Plot \(Beta\) node](#)
- [Working with Graph Output](#)
- [Exploring Graphs](#)
- [Editing Visualizations](#)
- [Fixing Errors](#)

Common Graph Nodes Features

Several phases of the data mining process use graphs and charts to explore data brought into IBM® SPSS® Modeler. For example, you can connect a Plot or Distribution node to a data source to gain insight into data types and distributions. You can then perform record and field

manipulations to prepare the data for downstream modeling operations. Another common use of graphs is to check the distribution and relationships between newly derived fields.

The Graphs palette contains the following nodes:

	The Graphboard node offers many different types of graphs in one single node. Using this node, you can choose the data fields you want to explore and then select a graph from those available for the selected data. The node automatically filters out any graph types that would not work with the field choices.
	The Plot node shows the relationship between numeric fields. You can create a plot by using points (a scatterplot) or lines.
	The Distribution node shows the occurrence of symbolic (categorical) values, such as mortgage type or gender. Typically, you might use the Distribution node to show imbalances in the data, which you could then rectify using a Balance node before creating a model.
	The Histogram node shows the occurrence of values for numeric fields. It is often used to explore the data before manipulations and model building. Similar to the Distribution node, the Histogram node frequently reveals imbalances in the data.
	The Collection node shows the distribution of values for one numeric field relative to the values of another. (It creates graphs that are similar to histograms.) It is useful for illustrating a variable or field whose values change over time. Using 3-D graphing, you can also include a symbolic axis displaying distributions by category.
	The Multiplot node creates a plot that displays multiple Y fields over a single X field. The Y fields are plotted as colored lines; each is equivalent to a Plot node with Style set to Line and X Mode set to Sort. Multiplots are useful when you want to explore the fluctuation of several variables over time.
	The Web node illustrates the strength of the relationship between values of two or more symbolic (categorical) fields. The graph uses lines of various widths to indicate connection strength. You might use a Web node, for example, to explore the relationship between the purchase of a set of items at an e-commerce site.
	The Time Plot node displays one or more sets of time series data. Typically, you would first use a Time Intervals node to create a <i>TimeLabel</i> field, which would be used to label the x axis.
	The Evaluation node helps to evaluate and compare predictive models. The evaluation chart shows how well models predict particular outcomes. It sorts records based on the predicted value and confidence of the prediction. It splits the records into groups of equal size (quantiles) and then plots the value of the business criterion for each quantile from highest to lowest. Multiple models are shown as separate lines in the plot.
	The Map Visualization node can accept multiple input connections and display geospatial data on a map as a series of layers. Each layer is a single geospatial field; for example, the base layer might be a map of a country, then above that you might have one layer for roads, one layer for rivers, and one layer for towns.
	The E-Plot (Beta) node shows the relationship between numeric fields. It is similar to the Plot node, but its options differ and its output uses a new graphing interface specific to this node. Use the beta-level node to play around with new graphing features.
	t-Distributed Stochastic Neighbor Embedding (t-SNE) is a tool for visualizing high-dimensional data. It converts affinities of data points to probabilities. This t-SNE node in SPSS Modeler is implemented in Python and requires the scikit-learn ® Python library.

When you have added a graph node to a stream, you can double-click the node to open a dialog box for specifying options. Most graphs contain a number of unique options presented on one or more tabs. There are also several tab options common to all graphs. The following topics contain more information about these common options.

When you have configured the options for a graph node, you can run it from within the dialog box or as part of a stream. In the generated graph window, you can generate Derive (Set and Flag) and Select nodes based on a selection or region of data, effectively "subsetting" the data. For example, you might use this powerful feature to identify and exclude outliers.

- [Aesthetics, Overlays, Panels, and Animation](#)
- [Using the Output Tab](#)
- [Using the Annotations Tab](#)
- [3-D Graphs](#)

Aesthetics, Overlays, Panels, and Animation

Overlays and Aesthetics

Aesthetics (and overlays) add dimensionality to a visualization. The effect of an aesthetic (grouping, clustering, or stacking) depends on the visualization type, the type of field (variable), and the graphic element type and statistic. For example, a categorical field for color may be used to group points in a scatterplot or to create the stacks in a stacked bar chart. A continuous numeric range for color may also be used to indicate the range's values for each point in a scatterplot.

You should experiment with the aesthetics and overlays to find one that fulfills your needs. The following descriptions may help you pick the right one.

Note: Not all aesthetics or overlays are available for all visualization types.

- **Color.** When color is defined by a categorical field, it splits the visualization based on the individual categories, one color for each category. When color is a continuous numeric range, it varies the color based on the value of the range field. If the graphic element (for example, a bar or box) represents more than one record/case and a range field is used for color, the color varies based on the *mean* of the range field.
- **Shape.** Shape is defined by a categorical field that splits the visualization into elements of different shapes, one for each category.
- **Transparency.** When transparency is defined by a categorical field, it splits the visualization based on the individual categories, one transparency level for each category. When transparency is a continuous numeric range, it varies the transparency based on the value of the range field. If the graphic element (for example, a bar or box) represents more than one record/case and a range field is used for transparency, the color varies based on the *mean* of the range field. At the largest value, the graphic elements are fully transparent. At the smallest value, they are fully opaque.
- **Data Label.** Data labels are defined by any type of field whose values are used to create labels that are attached to the graphic elements.
- **Size.** When size is defined by a categorical field, it splits the visualization based on the individual categories, one size for each category. When size is a continuous numeric range, it varies the size based on the value of the range field. If the graphic element (for example, a bar or box) represents more than one record/case and a range field is used for size, the size varies based on the *mean* of the range field.

Paneling and Animation

Paneling. Paneling, also known as faceting, creates a table of graphs. One graph is generated for each category in the paneling fields, but all panels appear simultaneously. Paneling is useful for checking whether the visualization is subject to the conditions of the paneling fields. For example, you may panel a histogram by gender to determine whether the frequency distributions are equal across males and females. That is, you can check whether salary is subject to gender differences. Select a categorical field for paneling.

Animation. Animation resembles paneling in that multiple graphs are created from the values of the animation field, but these graphs are not shown together. Rather, you use the controls in Explore mode to animate the output and flip through a sequence of individual graphs. Furthermore, unlike paneling, animation does not require a categorical field. You can specify a continuous field whose values are split up into ranges automatically. You can vary the size of the range with the animation controls in explore mode. Not all visualizations offer animation.

Related information

- [Common Graph Nodes Features](#)
 - [Using the Output Tab](#)
 - [Using the Annotations Tab](#)
 - [3-D Graphs](#)
-

Using the Output Tab

For all graph types, you can specify the following options for the filename and display of generated graphs.

Note: Distribution node graphs have additional settings.

Output name. Specifies the name of the graph produced when the node is run. Auto chooses a name based on the node that generates the output. Optionally, you can select Custom to specify a different name.

Output to screen. Select to generate and display the graph in a new window.

Output to file. Select to save the output as a file.

- Output Graph. Select to produce output in a graph format. Available only in Distribution nodes.
- Output Table. Select to produce output in a table format. Available only in Distribution nodes.
- Filename. Specify a filename used for the generated graph or table. Use the ellipsis button (...) to specify a specific file and location.
- File type. Specify the file type in the drop-down list. For all graph nodes, except the Distribution node with an Output Table option, the available graph file types are as follows.
 - Bitmap (.bmp)
 - PNG (.png)
 - Output object (.cou)
 - JPEG (.jpg)
 - HTML (.html)
 - ViZml document (.xml) for use in other IBM® SPSS® Statistics applications.

For the Output Table option in the Distribution node, the available file types are as follows.

- Tab delimited data (.tab)
- Comma delimited data (.csv)
- HTML (.html)
- Output object (.cou)

Paginate output. When saving output as HTML, this option is enabled to enable you to control the size of each HTML page. (Applies only to the Distribution node.)

Lines per page. When Paginate output is selected, this option is enabled to enable you to determine the length of each HTML page. The default setting is 400 rows. (Applies only to the Distribution node.)

Using the Annotations Tab

Used for all nodes, this tab offers options to rename nodes, supply a custom ToolTip, and store a lengthy annotation.

Related information

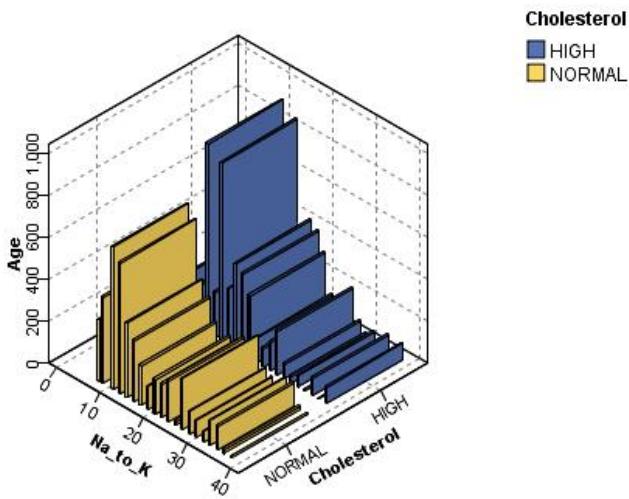
- [Common Graph Nodes Features](#)
- [Aesthetics, Overlays, Panels, and Animation](#)
- [Using the Output Tab](#)
- [3-D Graphs](#)

3-D Graphs

Plots and collection graphs in IBM® SPSS® Modeler have the ability to display information on a third axis. This provides you with additional flexibility in visualizing your data to select subsets or deriving new fields for modeling.

Once you have created a 3-D graph, you can click on it and drag your mouse to rotate it and view it from any angle.

Figure 1. Collection graph with x, y, and z axes



There are two ways of creating 3-D graphs in IBM SPSS Modeler: plotting information on a third axis (true 3-D graphs) and displaying graphs with 3-D effects. Both methods are available for plots and collections.

To Plot Information on a Third Axis

1. In the graph node dialog box, click the Plot tab.
2. Click the 3-D button to enable options for the z axis.
3. Use the Field Chooser button to select a field for the z axis. In some cases, only symbolic fields are allowed here. The Field Chooser will display the appropriate fields.

To Add 3-D Effects to a Graph

1. Once you have created a graph, click the Graph tab in the output window.
2. Click the 3-D button to switch the view to a three-dimensional graph.

Related information

- [Common Graph Nodes Features](#)
 - [Aesthetics, Overlays, Panels, and Animation](#)
 - [Using the Output Tab](#)
 - [Using the Annotations Tab](#)
-

Graphboard node

The Graphboard node enables you to choose from many different graphs outputs (bar charts, pie charts, histograms, scatterplots, heatmaps, etc.) in one single node. You begin, in the first tab, by choosing the data fields you want to explore, and then the node presents you with a choice of graph types that work for your data. The node automatically filters out any graph types that would not work with the field choices. You can define detailed, or more advanced graph options in the Detailed tab.

Note: You must connect the Graphboard node to a stream with data in order to edit the node or select graph types.

There are two buttons that enable you to control which visualization templates (and stylesheets, and maps) are available:

Manage. Manage visualization templates, stylesheets, and maps on your computer. You can import, export, rename, and delete visualization templates, stylesheets, and maps on your local machine. See the topic [Managing Templates, Stylesheets, and Map Files](#) for more information.

Location. Change the location in which visualization templates, stylesheets, and maps are stored. The current location is listed to the right of the button. See the topic [Setting the Location of Templates, Stylesheets, and Maps](#) for more information.

- [Graphboard Basic Tab](#)
 - [Graphboard Detailed Tab](#)
 - [Available Built-in Graphboard Visualization Types](#)
 - [Creating Map Visualizations](#)
 - [Graphboard Examples](#)
 - [Graphboard appearance tab](#)
 - [Setting the Location of Templates, Stylesheets, and Maps](#)
 - [Managing Templates, Stylesheets, and Map Files](#)
-

Graphboard Basic Tab

If you aren't sure which visualization type would best represent your data, use the Basic tab. When you select your data, you are then presented with a subset of visualization types that are appropriate for the data. For examples, see [Graphboard Examples](#).

1. Select one or more fields (variables) from the list. Use Ctrl+Click to select multiple fields.
Note that the measurement level of the field determines the type of visualizations that are available. You can change the measurement level by right-clicking the field in the list and choosing an option. For more information about the available measurement level types, see [Field \(Variable\) Types](#).
2. Select a visualization type. For descriptions of the available types, see [Available Built-in Graphboard Visualization Types](#).
3. For certain visualizations, you can choose a summary statistic. Different subsets of statistics are available depending on whether the statistic is count-based or calculated from a continuous field. The available statistics also depend on the template itself. A full list of statistics that may be available follows the next step.
4. If you want to define more options, such as optional aesthetics and panel fields, click Detailed. See the topic [Graphboard Detailed Tab](#) for more information.

Summary Statistics Calculated from a Continuous Field

- **Mean.** A measure of central tendency. The arithmetic average, the sum divided by the number of cases.
- **Median.** The value above and below which half of the cases fall, the 50th percentile. If there is an even number of cases, the median is the average of the two middle cases when they are sorted in ascending or descending order. The median is a measure of central tendency not sensitive to outlying values (unlike the mean, which can be affected by a few extremely high or low values).
- **Mode.** The most frequently occurring value. If several values share the greatest frequency of occurrence, each of them is a mode.
- **Minimum.** The smallest value of a numeric variable.
- **Maximum.** The largest value of a numeric variable.
- **Range.** The difference between the minimum and maximum values.
- **Mid Range.** The middle of the range, that is, the value whose difference from the minimum is equal to its difference from the maximum.

- **Sum.** The sum or total of the values, across all cases with nonmissing values.
- **Cumulative Sum.** The cumulative sum of the values. Each graphic element shows the sum for one subgroup plus the total sum of all previous groups.
- **Percent Sum.** The percentage within each subgroup based on a summed field compared to the sum across all groups.
- **Cumulative Percent Sum.** The cumulative percentage within each subgroup based on a summed field compared to the sum across all groups. Each graphic element shows the percentage for one subgroup plus the total percentage of all previous groups.
- **Variance.** A measure of dispersion around the mean, equal to the sum of squared deviations from the mean divided by one less than the number of cases. The variance is measured in units that are the square of those of the variable itself.
- **Standard Deviation.** A measure of dispersion around the mean. In a normal distribution, 68% of cases fall within one standard deviation of the mean and 95% of cases fall within two standard deviations. For example, if the mean age is 45, with a standard deviation of 10, 95% of the cases would be between 25 and 65 in a normal distribution.
- **Standard Error.** A measure of how much the value of a test statistic varies from sample to sample. It is the standard deviation of the sampling distribution for a statistic. For example, the standard error of the mean is the standard deviation of the sample means.
- **Kurtosis.** A measure of the extent to which there are outliers. For a normal distribution, the value of the kurtosis statistic is zero. Positive kurtosis indicates that the data exhibit more extreme outliers than a normal distribution. Negative kurtosis indicates that the data exhibit less extreme outliers than a normal distribution.
- **Skewness.** A measure of the asymmetry of a distribution. The normal distribution is symmetric and has a skewness value of 0. A distribution with a significant positive skewness has a long right tail. A distribution with a significant negative skewness has a long left tail. As a guideline, a skewness value more than twice its standard error is taken to indicate a departure from symmetry.

The following region statistics may result in more than one graphic element per subgroup. When using the interval, area, or edge graphic elements, a region statistic results in one graphic element showing the range. All other graphic elements result in two separate elements, one showing the start of the range and one showing the end of the range.

- **Region: Range.** The range of values between the minimum and maximum values.
- **Region: 95% Confidence Interval of Mean.** A range of values that has a 95% chance of including the population mean.
- **Region: 95% Confidence Interval of Individual.** A range of values that has a 95% chance of including the predicted value given the individual case.
- **Region: 1 Standard Deviation above/below Mean.** A range of values between 1 standard deviation above and below the mean.
- **Region: 1 Standard Error above/below Mean.** A range of values between 1 standard error above and below the mean.

Count-Based Summary Statistics

- **Count.** The number of rows/cases.
- **Cumulative Count.** The cumulative number of rows/cases. Each graphic element shows the count for one subgroup plus the total count of all previous groups.
- **Percent of Count.** The percentage of rows/cases in each subgroup compared to the total number of rows/cases.
- **Cumulative Percent of Count.** The cumulative percentage of rows/cases in each subgroup compared to the total number of rows/cases. Each graphic element shows the percentage for one subgroup plus the total percentage of all previous groups.
- **[Field \(Variable\) Types](#)**

Related information

- [Graphboard node](#)
- [Graphboard Detailed Tab](#)
- [Available Built-in Graphboard Visualization Types](#)
- [Creating Map Visualizations](#)
- [Graphboard Examples](#)

Field (Variable) Types

Icons appear next to fields in field lists to indicate the field type and data type. Icons also identify multiple response sets.

Table 1. Measurement level icons

Measurement Level	Numeric	String	Date	Time
Continuous		n/a		
Ordered Set				
Set				

Table 2. Multiple response set icons

Multiple response set type	Icon
Multiple response set, multiple categories	
Multiple response set, multiple dichotomies	

Measurement Level

A field's measurement level is important when you create a visualization. Following is a description of the measurement levels. You can temporarily change the measurement level by right-clicking a field in a field list and choosing an option. In most cases, you need to consider only the two broadest classifications of fields, categorical and continuous:

Categorical. Data with a limited number of distinct values or categories (for example, gender or religion). Categorical fields can be string (alphanumeric) or numeric fields that use numeric codes to represent categories (for example, 0 = *male* and 1 = *female*). Also referred to as qualitative data. Sets, ordered sets, and flags are all categorical fields.

- **Set.** A field/variable whose values represent categories with no intrinsic ranking (for example, the department of the company in which an employee works). Examples of nominal variables include region, zip code, and religious affiliation. Also known as a nominal variable.
- **Ordered Set.** A field/variable whose values represent categories with some intrinsic ranking (for example, levels of service satisfaction from highly dissatisfied to highly satisfied). Examples of ordered sets include attitude scores representing degree of satisfaction or confidence and preference rating scores. Also known as an ordinal variable.
- **Flag.** A field/variable with two distinct values, such as Yes and No or 1 and 2. Also known as a dichotomous or binary variable.

Continuous. Data measured on an interval or ratio scale, where the data values indicate both the order of values and the distance between values. For example, a salary of \$72,195 is higher than a salary of \$52,398, and the distance between the two values is \$19,797. Also referred to as quantitative, scale, or numeric range data.

Categorical fields define categories in the visualization, typically to draw separate graphic elements or to group graphic elements. Continuous fields are often summarized within categories of categorical fields. For example, a default visualization of income for gender categories would display the mean income for males and the mean income for females. The raw values for continuous fields can also be plotted, as in a scatterplot. For example, a scatterplot may show the current salary and beginning salary for each case. A categorical field could be used to group the cases by gender.

Data Types

Measurement level isn't the only property of a field that determines its type. A field is also stored as a specific data type. Possible data types are strings (non-numeric data such as letters), numeric values (real numbers), and dates. Unlike the measurement level, a field's data type cannot be changed temporarily. You must change the way the data are stored in the original data set.

Multiple Response Sets

Some data files support a special kind of "field" called a **multiple response set**. Multiple response sets aren't really "fields" in the normal sense. Multiple response sets use multiple fields to record responses to questions where the respondent can give more than one answer. Multiple response sets are treated like categorical fields, and most of the things you can do with categorical fields, you can also do with multiple response sets.

Multiple response sets can be multiple dichotomy sets or multiple category sets.

Multiple dichotomy sets. A multiple dichotomy set typically consists of multiple dichotomous fields: fields with only two possible values of a yes/no, present/absent, checked/not checked nature. Although the fields may not be strictly dichotomous, all of the fields in the set are coded the same way.

For example, a survey provides five possible responses to the question, "Which of the following sources do you rely on for news?" The respondent can indicate multiple choices by checking a box next to each choice. The five responses become five fields in the data file, coded 0 for No (not checked) and 1 for Yes (checked).

Multiple category sets. A multiple category set consists of multiple fields, all coded the same way, often with many possible response categories. For example, a survey item states, "Name up to three nationalities that best describe your ethnic heritage." There may be hundreds of possible responses, but for coding purposes the list is limited to the 40 most common nationalities, with everything else relegated to an "other" category. In the data file, the three choices become three fields, each with 41 categories (40 coded nationalities and one "other" category).

Graphboard Detailed Tab

Use the Detailed tab when you know what type of visualization you want to create or when you want to add optional aesthetics, panels, and/or animation to a visualization. For examples, see [Graphboard Examples](#).

1. If you selected a visualization type on the Basic tab, it will be displayed. Otherwise, choose one from the drop-down list. For information about the visualization types, see [Available Built-in Graphboard Visualization Types](#).
2. To the immediate right of the visualization's thumbnail image are controls for specifying the fields (variables) required for the visualization type. You must specify all of these fields.

3. For certain visualizations, you can select a summary statistic. In some cases (such as with bar charts), you can use one of these summary options for the transparency aesthetic. For descriptions of the summary statistics, see [Graphboard Basic Tab](#).
4. You can select one or more of the optional aesthetics. These can add dimensionality by allowing you to include other fields in the visualization. For example, you may use a field to vary the size of points in a scatterplot. For more information about optional aesthetics, see [Aesthetics, Overlays, Panels, and Animation](#). Please note that the transparency aesthetic is not supported through scripting.
5. If you are creating a map visualization, the Map Files group shows the map file or files that will be used. If there is a default map file, this file is displayed. To change the map file, click Select a Map File to display the Select Maps dialog box. You can also specify the default map file in this dialog box. See the topic [Selecting map files for map visualizations](#) for more information.
6. You can select one or more of the paneling or animation options. For more information about paneling and animation options, see [Aesthetics, Overlays, Panels, and Animation](#).

- [Selecting map files for map visualizations](#)

Related information

- [Graphboard node](#)
 - [Graphboard Basic Tab](#)
 - [Available Built-in Graphboard Visualization Types](#)
 - [Creating Map Visualizations](#)
 - [Graphboard Examples](#)
 - [Selecting map files for map visualizations](#)
-

Selecting map files for map visualizations

If you select a map visualization template, you need a map file that defines the geographic information for drawing the map. If there is a default map file, this will be used for the map visualization. To choose a different map file, click Select a Map File on the Detailed tab to display the Select Maps dialog box.

The Select Maps dialog box allows you to choose a primary map file and a reference map file. The map files define the geographic information for drawing the map. Your application is installed with a set of standard map files. If you have other ESRI shapefiles that you want to use, you first need to convert the shapefiles to SMZ files. See the topic [Converting and Distributing Map Shapefiles](#) for more information. After converting the map, click Manage... on the Template Chooser dialog box to import the map into the Manage system so that it is available in the Select Maps dialog box.

Following are some points to consider when specifying map files:

- All map templates require at least one map file.
- The map file typically links a map key attribute to the data key.
- If the template does not require a map key that links to a data key, it requires a reference map file and fields that specify coordinates (such as longitude and latitude) for drawing elements on the reference map.
- Overlay map templates require two maps: a primary map file and a reference map file. The reference map is drawn first so that it is behind the primary map file.

For information about map terminology such as attributes and features, see [Key Concepts for Maps](#).

Map File. You can select any map file that is in the Manage system. These include pre-installed map files and map files that you imported. For more information about managing map files, see [Managing Templates, Stylesheets, and Map Files](#).

Map Key. Specify the attribute that you want to use as the key that links the map file to the data key.

Save this map file and settings as the default. Select this checkbox if you want to use the selected map file as the default. If you have specified a default map file, you don't need to specify a map file every time you create a map visualization.

Data Key. This control lists the same value as appears on the Template Chooser Detailed tab. It is provided here for convenience if you need to change the key because of the specific map file that you choose.

Display all map features in the visualization. When this option is checked, all features in the map are rendered in the visualization, even if there is no matching data key value. If you want to see only the features for which you have data, clear this option. Features identified by the map keys shown in the Unmatched Map Keys list will not be rendered in the visualization.

Compare Map and Data Values. The map key and data key link to each other to create the map visualization. The map key and the data key should draw from the same domain (for example, countries and regions). Click Compare to test whether the data key and map key values match. The displayed icon informs you of the state of the comparison. These icons are described below. If a comparison has been performed and there are data key values with no matching map key values, the data key values appear in the Unmatched Data Keys list. In the Unmatched Map Keys list, you can also see which map key values have no matching data key values. If Display all map features in the visualization is not checked, features identified by these map key values will not be rendered.

Table 1. Comparison icons

Icon	Description
	No comparison has been performed. This is the default state before you click Compare. You should proceed with caution because you don't know if the data key and map key values match.
	A comparison has been performed, and the data key and map key values match completely. For every value of the data key, there is a matching feature identified by the map key.
	A comparison has been performed, and some data key and map key values do not match. For some data key values, there is no matching feature identified by the map key. You should proceed with caution. If you proceed, the map visualization will not include all data values.
	A comparison has been performed, and no data key and map key values match. You should choose a different data key or map key because no map will be rendered if you proceed.

Available Built-in Graphboard Visualization Types

You can create several different visualization types. All of the following built-in types are available on both the Basic and Detailed tabs. Some of the descriptions for the templates (especially the map templates) identify the fields (variables) specified on the Detailed tab using special text.

Table 1. Available graph types

Chart icon	Description	Chart icon	Description
	Bar Calculates a summary statistic for a continuous numeric field and displays the results for each category of a categorical field as bars. <i>Requires:</i> A categorical field and a continuous field.		Bar of Counts Displays the proportion of rows/cases in each category of a categorical field as bars. You can also use the Distribution graph node to produce this graph. That node offers some additional options. See the topic Distribution Node for more information. <i>Requires:</i> A single categorical field.
	Pie Calculates the sum of a continuous numeric field and displays the proportion of that sum distributed in each category of a categorical field as slices of a pie. <i>Requires:</i> A categorical field and a continuous field.		Pie of Counts Displays the proportion of rows/cases in each category of a categorical field as slices of a pie. <i>Requires:</i> A single categorical field.
	3-D Bar Calculates a summary statistic for a continuous numeric field and displays the results for the crossing of categories between two categorical fields. <i>Requires:</i> A pair of categorical fields and a continuous field.		3-D Pie This is the same as Pie except with an additional 3-D effect. <i>Requires:</i> A categorical field and a continuous field.
	Line Calculates a summary statistic for a field for each value of another field and draws a line connecting the values. You can also use the Plot graph node to produce a line plot graph. That node offers some additional options. See the topic Plot Node for more information. <i>Requires:</i> A pair of fields of any type.		Area Calculates a summary statistic for a field for each value of another field and draws an area connecting the values. The difference between a line and area is minimal in that an area resembles a line with the space colored below it. However, if you use a color aesthetic, this results in a simple split of the line and a stacking of the area. <i>Requires:</i> A pair of fields of any type.
	3-D Area Displays the values of one field plotted against the values of another and split by a categorical field. An area element is drawn for each category. <i>Requires:</i> A categorical field and a pair of fields of any type.		Path Displays the values of one field plotted against the values of another, with a line connecting the values in the order they appear in the original dataset. The ordering is the primary difference between a path and a line. <i>Requires:</i> A pair of fields of any type.
	Ribbon Calculates a summary statistic for a field for each value of another field and draws a ribbon connecting the values. A ribbon is essentially a line with 3-D effects. It is not a true 3-D graph. <i>Requires:</i> A pair of fields of any type.		Surface Displays the values of three fields plotted against the values of one another, with a surface connecting the values. <i>Requires:</i> Three fields of any type.

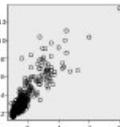
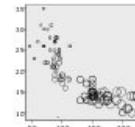
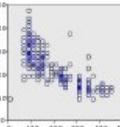
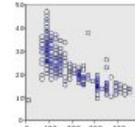
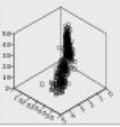
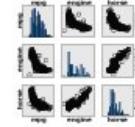
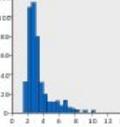
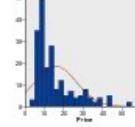
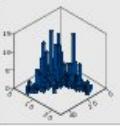
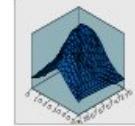
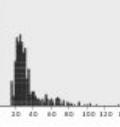
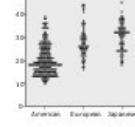
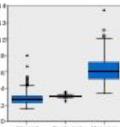
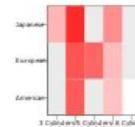
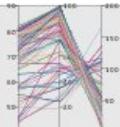
Chart icon	Description	Chart icon	Description
	<p>Scatterplot Displays the values of one field plotted against the values of another. This graph can highlight the relationship between the fields (if there is one). You can also use the Plot graph node to produce a scatterplot. That node offers some additional options. See the topic Plot Node for more information.</p> <p>Requires: A pair of fields of any type.</p>		<p>Bubble plot Like a basic scatterplot, displays the values of one field plotted against the values of another. The difference is that the values of a third field are used to vary the size of the individual points.</p> <p>Requires: Three fields of any type.</p>
	<p>Binned Scatterplot Like a basic scatterplot, displays the values of one field plotted against the values of another. The difference is that similar values are binned into groups and that the color or size aesthetic is used to indicate the number of cases in each bin.</p> <p>Requires : A pair of continuous fields.</p>		<p>Hex Binned Scatterplot See the description of Binned Scatterplot. The difference is the shape of the underlying bins, which are shaped like hexagons rather than circles. The resulting hex binned scatterplot looks similar to the binned scatterplot. However, the number of values in each bin will differ between the graphs because of the shape of the underlying bins.</p> <p>Requires: A pair of continuous fields.</p>
	<p>3-D Scatterplot Displays the values of three fields plotted against each other. This graph can highlight the relationship between the fields (if there is one). You can also use the Plot graph node to produce a 3-D scatterplot. That node offers some additional options. See the topic Plot Node for more information.</p> <p>Requires: Three fields of any type.</p>		<p>Scatterplot Matrix (SPLOM) Displays the values of one field plotted against the values of another for every field. A SPLOM is like a table of scatterplots. The SPLOM also includes a histogram of each field.</p> <p>Requires: Two or more continuous fields.</p>
	<p>Histogram Displays the frequency distribution of a field. A histogram can help you determine the distribution type and see whether the distribution is skewed. You can also use the Histogram graph node to produce this graph. That node offers some additional options. See the topic Histogram Plot Tab for more information.</p> <p>Requires: A single field of any type.</p>		<p>Histogram with Normal Distribution Displays the frequency distribution of a continuous field with a superimposed curve of the normal distribution.</p> <p>Requires: A single continuous field.</p>
	<p>3-D Histogram Displays the frequency distribution of a pair of continuous fields.</p> <p>Requires : A pair of continuous fields.</p>		<p>3-D Density Displays the frequency distribution of a pair of continuous fields. This is similar to a 3-D histogram, with the only difference being that a surface is used instead of bars to show the distribution.</p> <p>Requires: A pair of continuous fields.</p>
	<p>Dot plot Displays the individual cases/rows and stacks them at the distinct data points on the x axis. This graph is similar to a histogram in that it shows the distribution of the data, but it displays each case/row rather than an aggregated count for a specific bin (range of values).</p> <p>Requires : A single field of any type.</p>		<p>2-D Dot Plot Displays the individual cases/rows and stacks them at the distinct data points on the y axis for each category of a categorical field.</p> <p>Requires: A categorical field and a continuous field.</p>
	<p>Boxplot Calculates the five statistics (minimum, first quartile, median, third quartile, and maximum) for a continuous field for each category of a categorical field. The results are displayed as boxplot/schema elements. The boxplots can help you see how the distribution of continuous data varies among the categories.</p> <p>Requires: A categorical field and a continuous field.</p>		<p>Heat map Calculates the mean for a continuous field for the crossing of categories between two categorical fields.</p> <p>Requires : A pair of categorical fields and a continuous field.</p>
	<p>Parallel Creates parallel axes for each field and draws a line through the field value for every row/case in the data.</p> <p>Requires: Two or more continuous fields.</p>		<p>Choropleth of Counts Calculates the count for each category of a categorical field (Data Key) and draws a map that uses color saturation to represent the counts in the map features that correspond to the categories.</p> <p>Requires : A categorical field. A map file whose key matches the Data Key categories.</p>

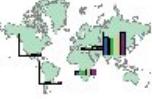
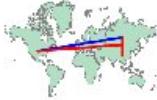
Chart icon	Description	Chart icon	Description
	Choropleth of Means/Medians/Sums Calculates the mean, median, or sum of a continuous field (Color) for each category of a categorical field (Data Key) and draws a map that uses color saturation to represent the calculated statistics in the map features that correspond to the categories. <i>Requires:</i> A categorical field and a continuous field. A map file whose key matches the Data Key categories.		Choropleth of Values Draws a map that uses color to represent values of a categorical field (Color) for the map features that correspond to values defined by another categorical field (Data Key). If there are multiple categorical values of the Color field for each feature, the modal value is used. <i>Requires:</i> A pair of categorical fields. A map file whose key matches the Data Key categories.
	Coordinates on a Choropleth of Counts This is similar to Choropleth of Counts except that there are two additional continuous fields (Longitude and Latitude) that identify coordinates for drawing points on the choropleth map. <i>Requires:</i> A categorical field and a pair of continuous fields. A map file whose key matches the Data Key categories.		Coordinates on a Choropleth of Means/Medians/Sums This is similar to Choropleth of Means/Medians/Sums except that there are two additional continuous fields (Longitude and Latitude) that identify coordinates for drawing points on the choropleth map. <i>Requires:</i> A categorical field and three continuous fields. A map file whose key matches the Data Key categories.
	Coordinates on a Choropleth of Values This is similar to Choropleth of Values except that there are two additional continuous fields (Longitude and Latitude) that identify coordinates for drawing points on the choropleth map. <i>Requires:</i> A pair of categorical fields and a pair of continuous fields. A map file whose key matches the Data Key categories.		Bars of Counts on a Map Calculates the proportion of rows/cases in each category of a categorical field (Categories) for each map feature (Data Key) and draws a map and the bar charts at the center of each map feature. <i>Requires:</i> A pair of categorical fields. A map file whose key matches the Data Key categories.
	Bars on a Map Calculates a summary statistic for a continuous field (Values) and displays the results for each category of a categorical field (Categories) for each map feature (Data Key) as bar charts positioned in the center of each map feature. <i>Requires:</i> A pair of categorical fields and a continuous field. A map file whose key matches the Data Key categories.		Pie of Counts of a Map Displays the proportion of rows/cases in each category of a categorical field (Categories) for each map feature (Data Key) and draws a map and the proportions as slices of a pie chart at the center of each map feature. <i>Requires:</i> A pair of categorical fields. A map file whose key matches the Data Key categories.
	Pie on a Map Calculates the sum of a continuous field (Values) in each category of a categorical field (Categories) for each map feature (Data Key) and draws a map and the sums as slices of a pie chart at the center of each map feature. <i>Requires:</i> A pair of categorical fields and a continuous field. A map file whose key matches the Data Key categories.		Line Chart on a Map Calculates a summary statistic for a continuous field (Y) for each value of another field (X) for each map feature (Data Key) and draws a map and line charts connecting the values at the center of each map feature. <i>Requires:</i> A categorical field and a pair of fields of any type. A map file whose key matches the Data Key categories.
	Coordinates on a Reference Map Draws a map and points using continuous fields (Longitude and Latitude) that identify coordinates for the points. <i>Requires:</i> A pair of range fields. A map file.		Arrows on a Reference Map Draws a map and arrows using continuous fields that identify the start points (Start Long and Start Lat) and end points (End Long and End Lat) for each arrow. Each record/case in the data results in an arrow in the map. <i>Requires:</i> Four continuous fields. A map file.
	Point Overlay Map Draws a reference map and superimposes another point map over it with point features colored by a categorical field (Color). <i>Requires:</i> A pair of categorical fields. A point map file whose key matches the Data Key categories. A reference map file.		Polygon Overlay Map Draws a reference map and superimposes another polygon map over it with polygon features colored by a categorical field (Color). <i>Requires:</i> A pair of categorical fields. A polygon map file whose key matches the Data Key categories. A reference map file.

Chart icon	Description	Chart icon	Description
	<p>Line Overlay Map Draws a reference map and superimposes another line map over it with line features colored by a categorical field (Color).</p> <p>Requires: A pair of categorical fields. A line map file whose key matches the Data Key categories. A reference map file.</p>		

Related information

- [Graphboard node](#)
- [Graphboard Basic Tab](#)
- [Graphboard Detailed Tab](#)
- [Creating Map Visualizations](#)
- [Graphboard Examples](#)
- [Graphboard appearance tab](#)

Creating Map Visualizations

For many visualizations, you have to make only two choices: fields (variables) of interest and a template to visualize those fields. No additional choices or actions are required. Map visualizations require at least one additional step: select a map file that defines the geographic information for the map visualization.

The basic steps for creating a simple map are the following:

1. Select the fields of interest on the Basic tab. For information about the type and number of fields required for different map visualizations, see [Available Built-in Graphboard Visualization Types](#).
2. Select a map template.
3. Click the Detailed tab.
4. Check that the Data Key and other required dropdown lists are set to the correct fields.
5. In the Map Files group, click Select a Map File.
6. Use the Select Maps dialog box to choose the map file and map key. The map key's values must match the values for of the field specified by Data Key. You can use the Compare button to compare these values. If you select an overlay map template, you will also need to choose a reference map. The reference map is not keyed to the data. It is used as the background for the main map. For more information about the Select Maps dialog box, see [Selecting map files for map visualizations](#).
7. Click OK to close the Select Maps dialog box.
8. In the Graphboard Template Chooser, click Run to create the map visualization.

Related information

- [Graphboard node](#)
- [Graphboard Basic Tab](#)
- [Graphboard Detailed Tab](#)
- [Available Built-in Graphboard Visualization Types](#)
- [Graphboard Examples](#)

Graphboard Examples

This section includes several different examples to demonstrate the available options. The examples also provide information for interpreting the resulting visualizations.

These examples use the stream named *graphboard.str*, which references the data files named *employee_data.sav*, *customer_subset.sav*, and *worldsales.sav*. These files are available from the *Demos* folder of any IBM® SPSS® Modeler Client installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *graphboard.str* file is in the *streams* folder.

It is recommended that you read the examples in the order presented. Subsequent examples build on previous ones.

- [Example: Bar Chart with a Summary Statistic](#)
- [Example: Stacked Bar Chart with a Summary Statistic](#)
- [Example: Paneled Histogram](#)
- [Example: Paneled Dot Plot](#)
- [Example: Boxplot](#)

- [Example: Pie Chart](#)
- [Example: Heat Map](#)
- [Example: Scatterplot Matrix \(SPLOM\)](#)
- [Example: Choropleth \(Color Map\) of Sums](#)
- [Example: Bar Charts on a Map](#)

Related information

- [Graphboard node](#)
- [Graphboard Basic Tab](#)
- [Graphboard Detailed Tab](#)
- [Available Built-in Graphboard Visualization Types](#)
- [Creating Map Visualizations](#)
- [Example: Bar Chart with a Summary Statistic](#)
- [Example: Stacked Bar Chart with a Summary Statistic](#)
- [Example: Paneled Histogram](#)
- [Example: Paneled Dot Plot](#)
- [Example: Boxplot](#)
- [Example: Pie Chart](#)
- [Example: Heat Map](#)
- [Example: Scatterplot Matrix \(SPLOM\)](#)
- [Example: Choropleth \(Color Map\) of Sums](#)
- [Example: Bar Charts on a Map](#)

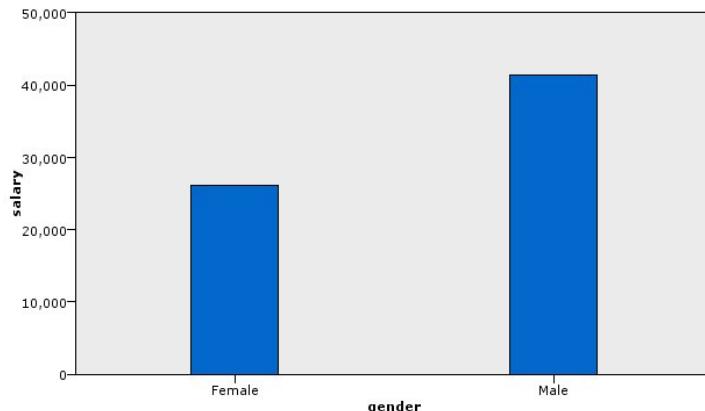
Example: Bar Chart with a Summary Statistic

We will create a bar chart that summarizes a continuous numeric field/variable for each category of a set/categorical variable. Specifically, we will create a bar chart that shows the mean salary for men and women.

This and several of the following examples use *Employee data*, which is a hypothetical dataset containing information about a company's employees.

1. Add a Statistics File source node that points to *employee_data.sav*.
2. Add a Graphboard node and open it for editing.
3. On the Basic tab, select *Gender* and *Current Salary*. (Use Ctrl+Click to select multiple fields/variables.)
4. Select Bar.
5. From the Summary drop-down list, select Mean.
6. Click Run.
7. On the resulting display, click the "Display field and value labels" toolbar button (the second of the group of two in the center of the toolbar).

Figure 1. Bar chart with a summary statistic



We can observe the following:

- Based on the height of the bars, it is clear that the mean salary for males is greater than the mean salary for females.

Related information

- [Graphboard Examples](#)

- [Example: Stacked Bar Chart with a Summary Statistic](#)
- [Example: Paneled Histogram](#)
- [Example: Paneled Dot Plot](#)
- [Example: Boxplot](#)
- [Example: Pie Chart](#)
- [Example: Heat Map](#)
- [Example: Scatterplot Matrix \(SPLOM\)](#)
- [Example: Choropleth \(Color Map\) of Sums](#)
- [Example: Bar Charts on a Map](#)

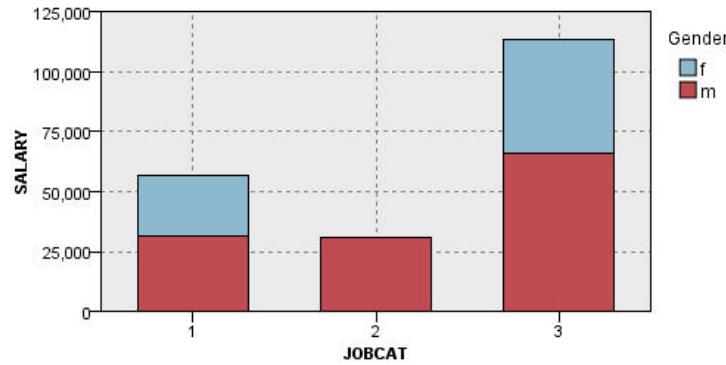
Example: Stacked Bar Chart with a Summary Statistic

We will now create a stacked bar chart to see whether the difference in mean salary between males and females is subject to job type. Perhaps females, on average, make more than males for certain job types.

Note: This example uses *Employee data*.

1. Add a Graphboard node and open it for editing.
2. On the Basic tab, select *Employment Category* and *Current Salary*. (Use Ctrl+Click to select multiple fields/variables.)
3. Select Bar.
4. From the Summary list, select Mean.
5. Click the Detailed tab. Note that your selections from the previous tab are reflected here.
6. In the Optional Aesthetics group, choose *gender* from the Color drop-down list.
7. Click Run.

Figure 1. Stacked bar chart



We can observe the following:

- The difference in mean salaries for each job type does not appear to be as great as it was in the bar chart that compared the mean salaries for all males and females. Perhaps there are varying number of males and females in each group. You could check this by creating a bar chart of counts.
- Regardless of job type, the mean salary for males is always greater than the mean salary for females.

Related information

- [Graphboard Examples](#)
- [Example: Bar Chart with a Summary Statistic](#)
- [Example: Paneled Histogram](#)
- [Example: Paneled Dot Plot](#)
- [Example: Boxplot](#)
- [Example: Pie Chart](#)
- [Example: Heat Map](#)
- [Example: Scatterplot Matrix \(SPLOM\)](#)
- [Example: Choropleth \(Color Map\) of Sums](#)
- [Example: Bar Charts on a Map](#)

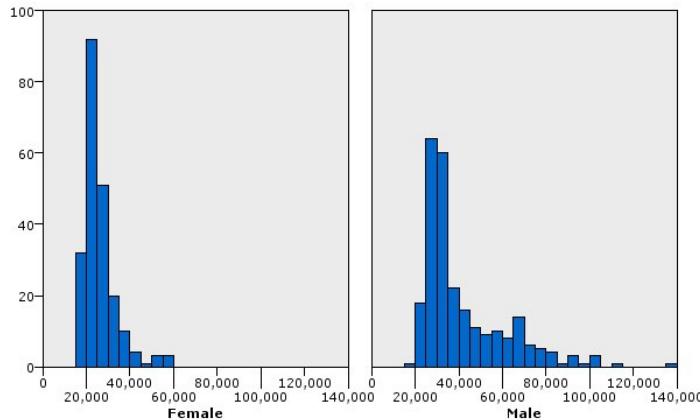
Example: Paneled Histogram

We will create a histogram that is paneled by gender so we can compare the frequency distributions of salary for men and women. The frequency distribution shows how many cases/rows lie within specific salary ranges. The paneled histogram can help us further analyze the difference in salaries between genders.

Note: This example uses *Employee data*.

1. Add a Graphboard node and open it for editing.
2. On the Basic tab, select *Current Salary*.
3. Select Histogram.
4. Click the Detailed tab.
5. In the Panels and Animation group, choose *gender* from the Panel Across drop-down list.
6. Click Run.

Figure 1. Paneled histogram



We can observe the following:

- Neither frequency distribution is a normal distribution. That is, the histograms do not resemble bell curves, as they would if the data were normally distributed.
- The taller bars are on the left side of each graph. Therefore, for both men and women, more make lower rather than higher salaries.
- The frequency distributions of salary among men and women are not equal. Notice the shape of the histograms. There are more men who make higher salaries than women who make higher salaries.

Related information

- [Graphboard Examples](#)
 - [Example: Bar Chart with a Summary Statistic](#)
 - [Example: Stacked Bar Chart with a Summary Statistic](#)
 - [Example: Paneled Dot Plot](#)
 - [Example: Boxplot](#)
 - [Example: Pie Chart](#)
 - [Example: Heat Map](#)
 - [Example: Scatterplot Matrix \(SPLOM\)](#)
 - [Example: Choropleth \(Color Map\) of Sums](#)
 - [Example: Bar Charts on a Map](#)
-

Example: Paneled Dot Plot

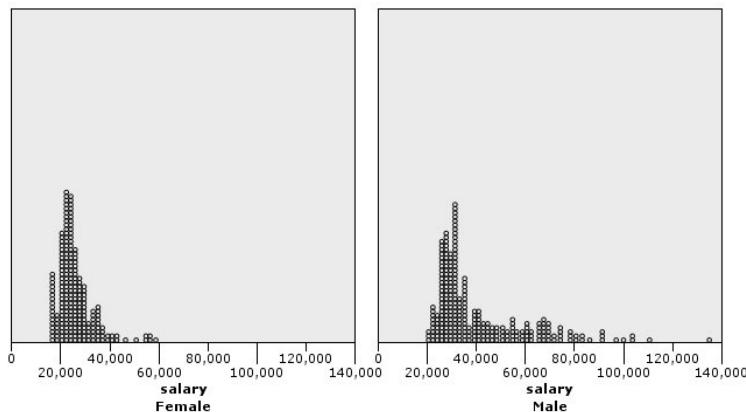
Like a histogram, a dot plot shows the distribution of a continuous numeric range. Unlike a histogram, which shows counts for binned ranges of data, a dot plot shows every row/case in the data. Therefore, a dot plot provides more granularity compared to the histogram. In fact, using a dot plot may be the preferred starting point when analyzing frequency distributions.

Note: This example uses *Employee data*.

1. Add a Graphboard node and open it for editing.
2. On the Basic tab, select *Current Salary*.
3. Select Dot Plot.
4. Click the Detailed tab.
5. In the Panels and Animation group, choose *gender* from the Panel Across drop-down list.
6. Click Run.

7. Maximize the resulting output window to view the plot more clearly.

Figure 1. Paneled dot plot



Compared to the histogram (see [Example: Paneled Histogram](#)), we can observe the following:

- The peak at 20,000 that appeared in the histogram for females is less dramatic in the dot plot. There are many cases/rows concentrated around that value, but most of those values are closer to 25,000. This level of granularity is not apparent in the histogram.
- Although the histogram for males suggests that the mean salary for males gradually declines after 40,000, the dot plot shows that the distribution is fairly uniform after that value, to 80,000. At any one salary value in that range, there are three or more males who earn that particular salary.

Related information

- [Graphboard Examples](#)
- [Example: Bar Chart with a Summary Statistic](#)
- [Example: Stacked Bar Chart with a Summary Statistic](#)
- [Example: Paneled Histogram](#)
- [Example: Boxplot](#)
- [Example: Pie Chart](#)
- [Example: Heat Map](#)
- [Example: Scatterplot Matrix \(SPLOM\)](#)
- [Example: Choropleth \(Color Map\) of Sums](#)
- [Example: Bar Charts on a Map](#)

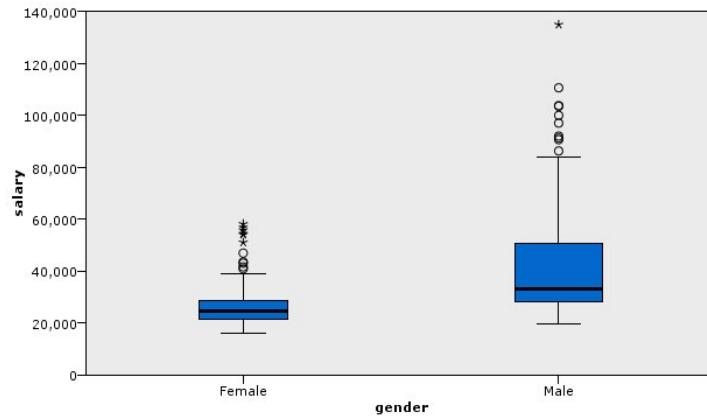
Example: Boxplot

A boxplot is another useful visualization for viewing how the data are distributed. A boxplot contains several statistical measures that we will explore after creating the visualization.

Note: This example uses *Employee data*.

1. Add a Graphboard node and open it for editing.
2. On the Basic tab, select *Gender* and *Current Salary*. (Use Ctrl+Click to select multiple fields/variables.)
3. Select Boxplot.
4. Click Run.

Figure 1. Boxplot



Let's explore the different parts of the boxplot:

- The dark line in the middle of the boxes is the median of *salary*. Half of the cases/rows have a value greater than the median, and half have a value lower. Like the mean, the median is a measure of central tendency. Unlike the mean, it is less influenced by cases/rows with extreme values. In this example, the median is lower than the mean (compare to [Example: Bar Chart with a Summary Statistic](#)). The difference between the mean and median indicates that there are a few cases/rows with extreme values that are elevating the mean. That is, there are a few employees who earn large salaries.
- The bottom of the box indicates the 25th percentile. Twenty-five percent of cases/rows have values below the 25th percentile. The top of the box represents the 75th percentile. Twenty-five percent of cases/rows have values above the 75th percentile. This means that 50% of the case/rows lie within the box. The box is much shorter for females than for males. This is one clue that *salary* varies less for females than for males. The top and bottom of the box are often called **hinges**.
- The T-bars that extend from the boxes are called **inner fences** or **whiskers**. These extend to 1.5 times the height of the box or, if no case/row has a value in that range, to the minimum or maximum values. If the data are distributed normally, approximately 95% of the data are expected to lie between the inner fences. In this example, the inner fences extend less for females compared to males, another indication that *salary* varies less for females than for males.
- The points are **outliers**. These are defined as values that do not fall in the inner fences. Outliers are extreme values. The asterisks or stars are **extreme outliers**. These represent cases/rows that have values more than three times the height of the boxes. There are several outliers for both females and males. Remember that the mean is greater than the median. The greater mean is caused by these outliers.

Related information

- [Graphboard Examples](#)
- [Example: Bar Chart with a Summary Statistic](#)
- [Example: Stacked Bar Chart with a Summary Statistic](#)
- [Example: Paneled Histogram](#)
- [Example: Paneled Dot Plot](#)
- [Example: Pie Chart](#)
- [Example: Heat Map](#)
- [Example: Scatterplot Matrix \(SPLOM\)](#)
- [Example: Choropleth \(Color Map\) of Sums](#)
- [Example: Bar Charts on a Map](#)

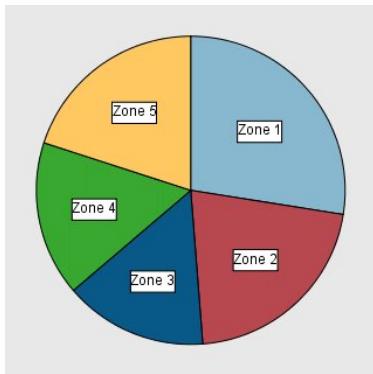
Example: Pie Chart

We will now use a different dataset to explore some other visualization types. The dataset is *customer_subset*, which is a hypothetical data file that contains information about customers.

We will first create a pie chart to check the proportion of customers in different geographic regions.

1. Add a Statistics File source node that points to *customer_subset.sav*.
2. Add a Graphboard node and open it for editing.
3. On the Basic tab, select *Geographic indicator*.
4. Select Pie of Counts.
5. Click Run.

Figure 1. Pie chart



We can observe the following:

- Zone 1 has more customers than in each of the other zones.
- Customers are equally distributed among the other zones.

Related information

- [Graphboard Examples](#)
 - [Example: Bar Chart with a Summary Statistic](#)
 - [Example: Stacked Bar Chart with a Summary Statistic](#)
 - [Example: Paneled Histogram](#)
 - [Example: Paneled Dot Plot](#)
 - [Example: Boxplot](#)
 - [Example: Heat Map](#)
 - [Example: Scatterplot Matrix \(SPLOM\)](#)
 - [Example: Choropleth \(Color Map\) of Sums](#)
 - [Example: Bar Charts on a Map](#)
-

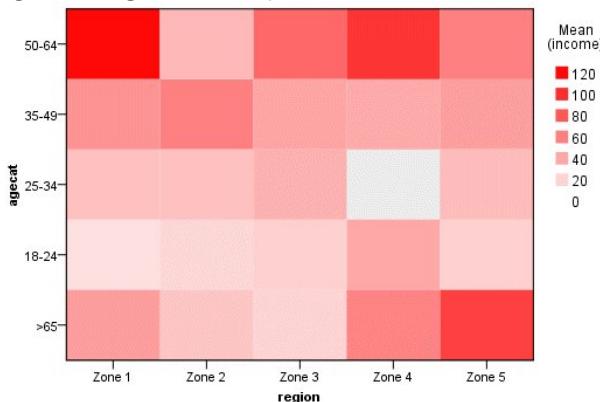
Example: Heat Map

We will now create a categorical heat map to check the mean income for customers in different geographic regions and age groups.

Note: This example uses *customer_subset*.

1. Add a Graphboard node and open it for editing.
2. On the Basic tab, select *Geographic indicator*, *Age category*, and *Household income in thousands*, in that order. (Use Ctrl+Click to select multiple fields/variables.)
3. Select Heat Map.
4. Click Run.
5. On the resulting output window, click the "Display field and value labels" toolbar button (the right-hand one of the two in the center of the toolbar).

Figure 1. Categorical heat map



We can observe the following:

- A heat map is like a table that uses colors instead of numbers to represent the values for the cells. Bright, deep red indicates the highest value, while gray indicates a low value. The value of each cell is the mean of the continuous field/variable for each pair of categories.

- Except in Zone 2 and Zone 5, the group of customers whose age is between 50 and 64 have a greater mean household income than those in other groups.
- There are no customers between the ages of 25 and 34 in Zone 4.

Related information

- [Graphboard Examples](#)
 - [Example: Bar Chart with a Summary Statistic](#)
 - [Example: Stacked Bar Chart with a Summary Statistic](#)
 - [Example: Paneled Histogram](#)
 - [Example: Paneled Dot Plot](#)
 - [Example: Boxplot](#)
 - [Example: Pie Chart](#)
 - [Example: Scatterplot Matrix \(SPLOM\)](#)
 - [Example: Choropleth \(Color Map\) of Sums](#)
 - [Example: Bar Charts on a Map](#)
-

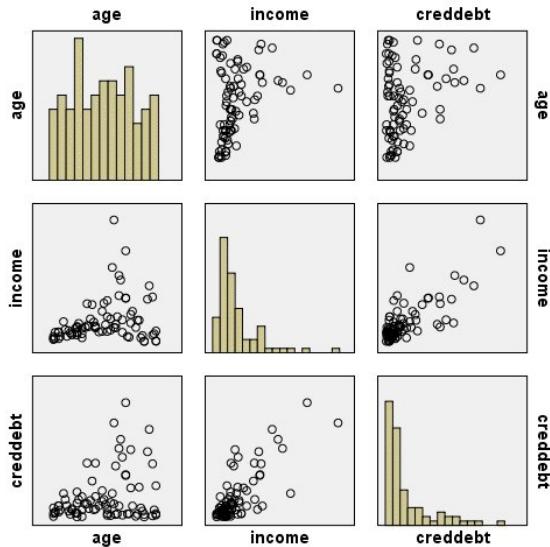
Example: Scatterplot Matrix (SPLOM)

We will create a scatterplot matrix of several different variables so that we can determine whether there are any relationships among the variables in the dataset.

Note: This example uses *customer_subset*.

1. Add a Graphboard node and open it for editing.
2. On the Basic tab, select *Age in years*, *Household income in thousands*, and *Credit card debt in thousands*. (Use Ctrl+Click to select multiple fields/variables.)
3. Select SPLOM.
4. Click Run.
5. Maximize the output window to view the matrix more clearly.

Figure 1. Scatterplot matrix (SPLOM)



We can observe the following:

- The histograms displayed on the diagonal show the distribution of each variable in the SPLOM. The histogram for *age* appears in the upper left cell, that for *income* in the center cell, and that for *creddebt* in the lower right cell. None of the variables appears normally distributed. That is, none of the histograms resembles a bell curve. Also, note that the histograms for *income* and *creddebt* are positively skewed.
- There does not seem to be any relationship between *age* and the other variables.
- There is a linear relationship between *income* and *creddebt*. That is, *creddebt* increases as *income* increases. You may want to create individual scatterplots of these variables and the other related variables to explore the relationships further.

Related information

- [Graphboard Examples](#)
- [Example: Bar Chart with a Summary Statistic](#)

- [Example: Stacked Bar Chart with a Summary Statistic](#)
- [Example: Paneled Histogram](#)
- [Example: Paneled Dot Plot](#)
- [Example: Boxplot](#)
- [Example: Pie Chart](#)
- [Example: Heat Map](#)
- [Example: Choropleth \(Color Map\) of Sums](#)
- [Example: Bar Charts on a Map](#)

Example: Choropleth (Color Map) of Sums

We will now create a map visualization. Then, in the subsequent example, we will create a variation of this visualization. The dataset is *worldsales*, which is a hypothetical data file that contains sales revenue by continent and product.

1. Add a Graphboard node and open it for editing.
2. On the Basic tab, select *Continent* and *Revenue*. (Use Ctrl+Click to select multiple fields/variables.)
3. Select Choropleth of Sums.
4. Click the Detailed tab.
5. In the Optional Aesthetics group, choose *Continent* from the Data Label drop-down list.
6. In the Map Files group, click Select a Map File.
7. In the Select Maps dialog box, check that Map is set to *Continents* and Map Key is set to *CONTINENT*.
8. In the Compare Map and Data Values groups, click Compare to ensure the map keys match the data keys. In this example, all of data key values have matching map keys and features. We can also see that there is no data for Oceania.
9. In the Select Maps dialog box, click OK.
10. Click Run.

Figure 1. Choropleth of Sums



From that map visualization, we can easily see that revenue is highest in North America and lowest in South America and Africa. Each continent is labeled because we used *Continent* for the data label aesthetic.

Related information

- [Graphboard Examples](#)
- [Example: Bar Chart with a Summary Statistic](#)
- [Example: Stacked Bar Chart with a Summary Statistic](#)
- [Example: Paneled Histogram](#)
- [Example: Paneled Dot Plot](#)
- [Example: Boxplot](#)
- [Example: Pie Chart](#)
- [Example: Heat Map](#)
- [Example: Scatterplot Matrix \(SPLOM\)](#)
- [Example: Bar Charts on a Map](#)

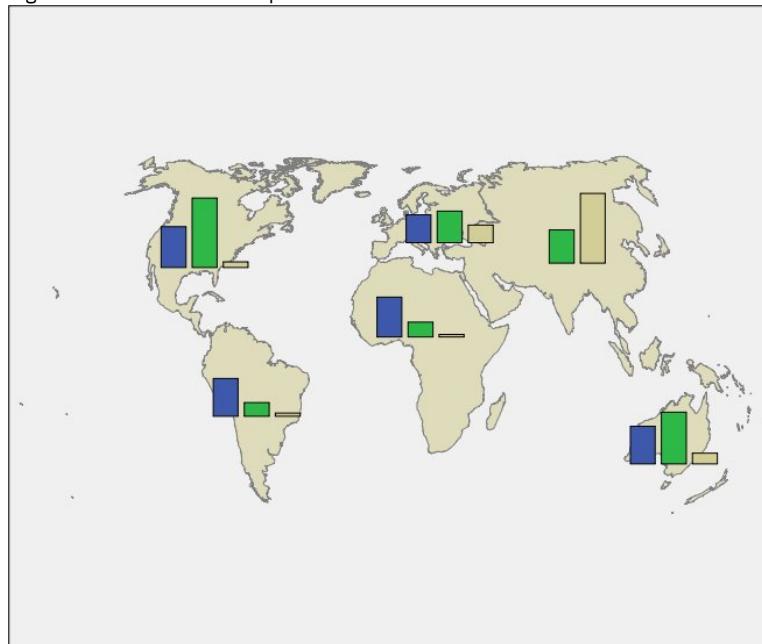
Example: Bar Charts on a Map

This example shows how revenue breaks down by product in each continent.

Note: This example uses *worlsales*.

1. Add a Graphboard node and open it for editing.
2. On the Basic tab, select *Continent*, *Product*, and *Revenue*. (Use Ctrl+Click to select multiple fields/variables.)
3. Select Bars on a Map.
4. Click the Detailed tab.
When using more than one field of a specific type, it is important to check that each field is assigned to the correct slot.
5. From the Categories drop-down list, choose *Product*.
6. From the Values drop-down list, choose *Revenue*.
7. From the Data Key drop-down list, choose *Continent*.
8. From the Summary drop-down list, choose *Sum*.
9. In the Map Files group, click Select a Map File.
10. In the Select Maps dialog box, check that Map is set to *Continents* and Map Key is set to *CONTINENT*.
11. In the Compare Map and Data Values groups, click Compare to ensure the map keys match the data keys. In this example, all of data key values have matching map keys and features. We can also see that there is no data for Oceania.
12. On the Select Maps dialog box, click OK.
13. Click Run.
14. Maximize the resulting output window to see the display more clearly.

Figure 1. Bar Charts on a Map



We can observe the following:

- The distribution of total revenue across products is very similar in South America and Africa.
- *Product C* generates the least revenue everywhere except in Asia.
- There is no or minimal revenue from *Product A* in Asia.

Related information

- [Graphboard Examples](#)
- [Example: Bar Chart with a Summary Statistic](#)
- [Example: Stacked Bar Chart with a Summary Statistic](#)
- [Example: Paneled Histogram](#)
- [Example: Paneled Dot Plot](#)
- [Example: Boxplot](#)
- [Example: Pie Chart](#)
- [Example: Heat Map](#)
- [Example: Scatterplot Matrix \(SPLOM\)](#)
- [Example: Choropleth \(Color Map\) of Sums](#)

Graphboard appearance tab

You can specify appearance options before graph creation.

General appearance options

Title. Enter the text to use for the graph's title.

Subtitle. Enter the text to use for the graph's subtitle.

Caption. Enter the text to use for the graph's caption.

Sampling. Specify a method for larger datasets. You can specify a maximum dataset size or use the default number of records. Performance is enhanced for large datasets when you select the Sample option. Alternatively, you can choose to plot all data points by selecting Use all data, but you should note that this may dramatically decrease the performance of the software.

Stylesheet appearance options

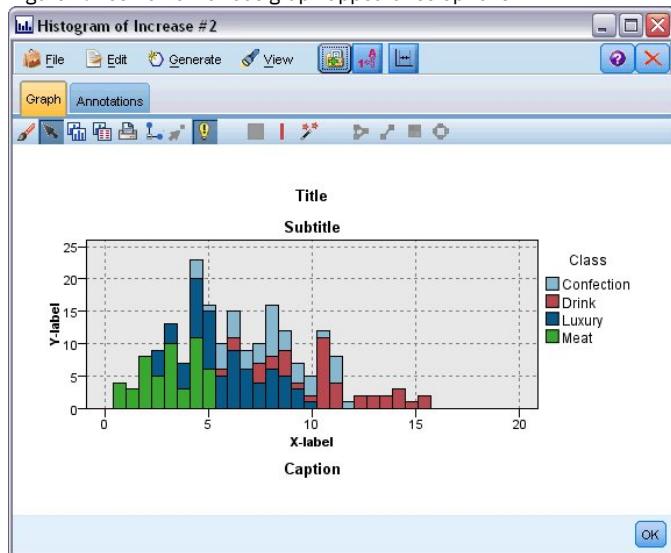
There are two buttons that enable you to control which visualization templates (and stylesheets, and maps) are available:

Manage. Manage visualization templates, stylesheets, and maps on your computer. You can import, export, rename, and delete visualization templates, stylesheets, and maps on your local machine. See the topic [Managing Templates, Stylesheets, and Map Files](#) for more information.

Location. Change the location in which visualization templates, stylesheets, and maps are stored. The current location is listed to the right of the button. See the topic [Setting the Location of Templates, Stylesheets, and Maps](#) for more information.

The following example shows where appearance options are placed on a graph. (Note that not all graphs use all these options.)

Figure 1. Position of various graph appearance options



Setting the Location of Templates, Stylesheets, and Maps

Visualization templates, visualization stylesheets, and map files are stored in a specific local folder or in the IBM® SPSS® Collaboration and Deployment Services Repository. When selecting templates, stylesheets, and maps, only the built-in ones in this location are displayed. By keeping all templates, stylesheets, and map files in one place, IBM SPSS applications can easily access them. For information about adding additional templates, stylesheets, and map files to this location, see [Managing Templates, Stylesheets, and Map Files](#).

How to Set the Location of Templates, Stylesheets, and Map Files

1. In a template or stylesheet dialog box, click Location... to display the Templates, Stylesheets, and Maps dialog box.
2. Select an option for the default location for templates, stylesheets, and map files:

Local Machine. Templates, stylesheets, and map files are located in a specific folder on your local computer. On Windows XP, this folder is C:\Documents and Settings\<user>\Application Data\SPSSInc\Graphboard. The folder cannot be changed.

IBM SPSS Collaboration and Deployment Services Repository. Templates, stylesheets, and map files are located in a user-specified folder in the IBM SPSS Collaboration and Deployment Services Repository. To identify the specific folder, click Folder. For more information, see [Using the IBM SPSS Collaboration and Deployment Services Repository as the Template, Stylesheet, and Map File Location](#).

3. Click OK.

- [Using the IBM SPSS Collaboration and Deployment Services Repository as the Template, Stylesheet, and Map File Location](#)
-

Using the IBM SPSS Collaboration and Deployment Services Repository as the Template, Stylesheet, and Map File Location

Visualization templates and stylesheets can be stored in the IBM® SPSS® Collaboration and Deployment Services Repository. This location is a specific folder in the IBM SPSS Collaboration and Deployment Services Repository. If this is set as the default location, any templates, stylesheets, and map files in this location are available for selection.

This feature requires the Statistics Adapter option.

How to Set a Folder in IBM SPSS Collaboration and Deployment Services Repository as the Location for Templates, Stylesheets, and Map Files

1. In a dialog box with a Location button, click Location....
2. Select IBM SPSS Collaboration and Deployment Services Repository.
3. Click Folder.

Note: If you are not already connected to the IBM SPSS Collaboration and Deployment Services Repository, you will be prompted for connection information.

4. In the Select Folder dialog box, select the folder in which templates, stylesheets, and map files are stored.
5. Optionally, you can select a label from Retrieve Label. Only templates, stylesheets, and map files with that label will be displayed.
6. If you are looking for a folder that contains a particular template or stylesheet, you may want to search for the template, stylesheet, or map file on the Search tab. The Select Folder dialog box automatically selects the folder in which the found template, stylesheet, or map file is located.
7. Click Select Folder.

Managing Templates, Stylesheets, and Map Files

You can manage the templates, stylesheets, and map files in the local location on your computer by using the Manage Templates, Stylesheets, and Maps dialog box. This dialog box allows you to import, export, rename, and delete visualization templates, stylesheets, and map files in the local location on your computer.

Click Manage... in one of the dialog boxes where you select templates, stylesheets, or maps.

Manage Templates, Stylesheets, and Maps Dialog Box

The Template tab lists all the local templates. The Stylesheet tab lists all the local stylesheets, as well as displaying example visualizations with sample data. You can select one of the stylesheets to apply its styles to the example visualizations. See the topic [Applying stylesheets](#) for more information. The Map tab lists all the local map files. This tab also displays the map keys including sample values, a comment if one was provided when creating the map, and a preview of the map.

The following buttons operate on whichever tab is currently activated.

Import. Import a visualization template, stylesheet, or map file from the file system. Importing a template, stylesheet, or map file makes it available to the IBM® SPSS® application. If another user sent you a template, stylesheet, or map file, you import the file before using it in your application.

Export. Export a visualization template, stylesheet, or map file to the file system. Export a template, stylesheet, or map file when you want to send it to another user.

Rename. Rename the selected visualization template, stylesheet, or map file. You cannot change a name to one that's already used.

Export Map Key. Export the map keys as a comma-separated values (CSV) file. This button is enabled only on the Map tab.

Delete. Delete the selected visualization template(s), stylesheet(s), or map file(s). You can select multiple templates, stylesheets, or map files with Ctrl-click. There is no undo action for deleting so proceed with caution.

Converting and Distributing Map Shapefiles

The Graphboard Template Chooser allows you to create map visualizations from the combination of a visualization template and an SMZ file. SMZ files are similar to ESRI shapefiles (SHP file format) in that they contain the geographic information for drawing a map (for example, country borders), but they are optimized for map visualizations. The Graphboard Template Chooser is pre-installed with a select number of SMZ files. If you have an existing ESRI shapefile that you want to use for map visualizations, you first need to convert the shapefile to an SMZ file using the Map Conversion Utility. The Map Conversion Utility supports point, polyline, or polygon (shape types 1, 3, and 5) ESRI shapefiles containing a single layer.

In addition to converting ESRI shapefiles, the Map Conversion Utility allows you to modify the map's level of detail, change feature labels, merge features, and move features, among many other optional changes. You can also use the Map Conversion Utility to modify an existing SMZ file (including the pre-installed ones).

Editing pre-installed SMZ files

1. Export the SMZ file from the Manage system. See the topic [Managing Templates, Stylesheets, and Map Files](#) for more information.
2. Use the Map Conversion Utility to open and edit the exported SMZ file. It is recommended that you save the file with a different name. See the topic [Using the Map Conversion Utility](#) for more information.
3. Import the modified SMZ file into the Manage system. See the topic [Managing Templates, Stylesheets, and Map Files](#) for more information.

Additional Resources for Map Files

Geospatial data in the SHP file format, which could be used to support your mapping needs, is available from many private and public sources. Check with local government web sites if you seek free data. Many of the templates in this product are based on publicly available data obtained from GeoCommons (<http://www.geocommons.com>) and the U.S. Census Bureau (<http://www.census.gov>).

IMPORTANT NOTICE: Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM program, unless indicated in a Notices file accompanying this IBM program, and use of the materials sites is at your own risk.

- [Key Concepts for Maps](#)
- [Using the Map Conversion Utility](#)
- [Distributing map files](#)

Key Concepts for Maps

Understanding some key concepts related to shapefiles will help you use the Map Conversion Utility effectively.

A **shapefile** provides the geographic information for drawing a map. There are three types of shapefiles that the Map Conversion Utility supports:

- **Point.** The shapefile identifies the locations of points, such as cities.
- **Polyline.** The shapefile identifies paths and their locations, such as rivers.
- **Polygon.** The shapefile identifies bounded regions and their locations, such as countries.

Most commonly, you will use a polygon shapefile. Choropleth maps are created from polygon shapefiles. Choropleth maps use color to represent a value within individual polygons (regions). Point and polyline shapefiles are typically overlaid on a polygon shapefile. An example is a point shapefile of U.S. cities overlaid on a polygon shapefile of U.S. states.

A shapefile is comprised of **features**. Features are the individual geographical entities. For example, features may be countries, states, cities, and so on. The shapefile also contains data about the features. These data are stored in **attributes**. Attributes are similar to fields or variables in a data file. There is at least one attribute that is the **map key** for the feature. The map key may be a label, such as country or state name. The map key is what you will link to a variable/field in a data file to create a map visualization.

Note that you will be able to retain only the key attribute or attributes in the SMZ file. The Map Conversion Utility does not support saving additional attributes. This means that you will need to create multiple SMZ files if you want to aggregate at different levels. For example, if you want to aggregate U.S. states and regions, you will need separate SMZ files: one that has a key identifying states and one that has a key identifying regions.

Related information

- [Converting and Distributing Map Shapefiles](#)
- [Using the Map Conversion Utility](#)
- [Distributing map files](#)

Using the Map Conversion Utility

How to start the Map Conversion Utility

From the menus choose:

Tools > Map Conversion Utility

There are four major screens (steps) in the Map Conversion Utility. One of the steps also involves sub-steps for more detailed control over editing the map file.

- [Step 1 - Choose Destination and Source Files](#)
- [Step 2 - Choose Map Key](#)
- [Step 3 - Edit the Map](#)
- [Step 4 - Finish](#)

Related information

- [Converting and Distributing Map Shapefiles](#)
- [Key Concepts for Maps](#)
- [Distributing map files](#)

Step 1 - Choose Destination and Source Files

You first have to select a source map file and a destination for the converted map file. You will need both the .shp and .dbf files for the shapefile.

Select a .shp (ESRI) or .smz file for conversion. Browse to an existing map file on your computer. This is the file that you will convert to and save as an SMZ file. The .dbf file for the shapefile *must* be stored in the same location with a base file name that matches the .shp file. The .dbf file is required because it contains the attribute information for the .shp file.

Set a destination and file name for the converted map file. Enter a path and file name for the SMZ file that will be created from the original map source.

- **Import to the Template Chooser.** In addition to saving a file on the file system, you can optionally add the map to the Manage list of the Template Chooser. If you choose this option, the map will automatically be available in the Template Chooser for IBM® SPSS® products installed on your computer. If you don't import to the Template Chooser now, you will need to import it manually at a later time. For more information about importing maps into the Template Chooser Manage system, see [Managing Templates, Stylesheets, and Map Files](#).

Related information

- [Using the Map Conversion Utility](#)
- [Step 2 - Choose Map Key](#)
- [Step 3 - Edit the Map](#)
- [Step 4 - Finish](#)

Step 2 - Choose Map Key

Now you will choose which map keys to include in the SMZ file. You can then change some options that will affect rendering of the map. The subsequent steps in the Map Conversion Utility include a preview of the map. The rendering options that you choose will be used to generate the map preview.

Choose primary map key. Select the attribute that is the primary key for identifying and labeling features in the map. For example, a world map's primary key may be the attribute identifying country names. The primary key will also link your data to the map features, so be sure that the values (labels) of the attribute you choose will match the values in your data. Example labels are displayed when you choose an attribute. If you need to change these labels, you will be able to do so in a later step.

Choose additional keys to include. In addition to the primary map key, check any other key attributes that you want to include in the generated SMZ file. For example, some attributes may contain translated labels. If you expect data coded in other languages, you may want to preserve these attributes. Note that you can choose only those additional keys that represent the same features as the primary key. For example if the primary key were the full names of U.S. states, you can select only those alternate keys that represent U.S. States, such as state abbreviations.

Automatically smooth the map. Shapefiles with polygons typically contain too many data points and too much detail for statistical map visualizations. The excess details can be distracting and negatively impact performance. You can reduce the level of detail and generalize the map with smoothing. The map will look crisper and render more quickly as a result. When the map is automatically smoothed, the maximum angle is 15 degrees and the percentage to keep is 99. For information about these settings, see [Smooth the map](#). Note that you have the opportunity to apply additional smoothing later in another step.

Remove boundaries between touching polygons in the same feature. Some features may include sub-features that have boundaries internal to the main features of interest. For example, a world map of continents may contain internal boundaries for the countries contained within each continent. If you choose this option, the internal boundaries will not appear in the map. In the world map of continents example, choosing this option would remove the country borders, while retaining the continent borders.

Related information

- [Using the Map Conversion Utility](#)
 - [Step 1 - Choose Destination and Source Files](#)
 - [Step 3 - Edit the Map](#)
 - [Step 4 - Finish](#)
-

Step 3 - Edit the Map

Now that you have specified basic options for the map, you can edit more specific ones. These modifications are optional. This step of the Map Conversion Utility guides you through the associated tasks and displays a preview of the map so you can verify your changes. Some tasks may not be available depending on shapefile type (point, polyline, or polygon) and coordinate system.

Every task has the following common controls on the left side of the Map Conversion Utility.

Show the labels on the map. By default, feature labels are not shown in the preview. You can choose to display the labels. Although the labels can help identify features, they may interfere with direct selection on the preview map. Turn on this option when you need it, for example when you are editing the feature labels.

Color the map preview. By default, the preview map displays areas with a solid color. All features are the same color. You can choose to have an assortment of colors assigned to the individual map features. This option may help distinguish different features in the map. It will be especially helpful when you merge features and want to see how the new features are represented in the preview.

Every task also has the following common control on the right side of the Map Conversion Utility.

Undo. Click Undo to revert to the last state. You can undo a maximum of 100 changes.

- [Smooth the map](#)
- [Edit the feature labels](#)
- [Merge features](#)
- [Move features](#)
- [Delete features](#)
- [Delete individual elements](#)
- [Set the projection](#)

Related information

- [Using the Map Conversion Utility](#)
 - [Step 1 - Choose Destination and Source Files](#)
 - [Step 2 - Choose Map Key](#)
 - [Step 4 - Finish](#)
-

Smooth the map

Shapefiles with polygons typically contain too many data points and too much detail for statistical map visualizations. The excess details can be distracting and negatively impact performance. You can reduce the level of detail and generalize the map with smoothing. The map will look crisper and render more quickly as a result. This option is not available for point and polyline maps.

Max. angle. The maximum angle, which must be a value between 1 and 20, specifies the tolerance for smoothing sets of points that are almost linear. A larger value allows more tolerance for the linear smoothing and will subsequently drop more points, resulting in a more generalized map. To apply linear smoothing, the Map Conversion Utility checks the inner angle formed by every set of three points in the map. If 180 minus the angle is less than the specified value, the Map Conversion Utility will drop the middle point. In other words, the Map Conversion Utility is checking

whether the line formed by the three points is almost straight. If it is, the Map Conversion Utility treats the line as a straight line between the endpoints and drops the middle point.

Percent to keep. The percentage to keep, which must be a value between 90 and 100, determines the amount of land area to keep when the map is smoothed. This option affects only those features that have multiple polygons, as would be the case if a feature included islands. If the total area of a feature minus a polygon is greater than the specified percentage of the original area, the Map Conversion Utility will drop the polygon from the map. The Map Conversion Utility will never remove all polygons for the feature. That is, there will always be at least one polygon for the feature, regardless of the amount of smoothing applied.

After you choose a maximum angle and percentage to keep, click Apply. The preview will update with the smoothing changes. If you need to smooth the map again, repeat until you obtain the level of smoothness you want. Note that there is a limit to the smoothing. If you repeatedly smooth, you will reach a point when no additional smoothing can be applied to the map.

Related information

- [Step 3 - Edit the Map](#)
 - [Edit the feature labels](#)
 - [Merge features](#)
 - [Move features](#)
 - [Delete features](#)
 - [Delete individual elements](#)
 - [Set the projection](#)
-

Edit the feature labels

You can edit the feature labels as needed (perhaps to match your expected data) and also re-position the labels in the map. Even if you don't think you need to change the labels, you should review them before creating visualizations from the map. Because labels are not shown by default in the preview, you may also want to select Show the labels on the map to display them.

Keys. Select the key containing the feature labels that you want to review and/or edit.

Features. This list displays the feature labels contained in the selected key. To edit the label, double-click it in the list. If labels are shown on the map, you can also double-click the feature labels directly in the map preview. If you want to compare the labels to an actual data file, click Compare.

X/Y. These text boxes list the current center point of the selected feature's label on the map. The units are displayed in the map's coordinates. These may be local, Cartesian coordinates (for example, the State Plane Coordinate System) or geographic coordinates (where X is longitude and Y is latitude). Enter coordinates for the new position of the label. If labels are shown, you can also click and drag a label on the map to move it. The text boxes will update with the new position.

Compare. If you have a data file that contains data values that are intended to match the features labels for a particular key, click Compare to display the Compare to an External Data Source dialog box. In this dialog box you will be able to open the data file and compare its values directly with those in the map key's feature labels.

- [Compare to an External Data Source dialog box](#)

Related information

- [Step 3 - Edit the Map](#)
 - [Smooth the map](#)
 - [Merge features](#)
 - [Move features](#)
 - [Delete features](#)
 - [Delete individual elements](#)
 - [Set the projection](#)
 - [Compare to an External Data Source dialog box](#)
-

Compare to an External Data Source dialog box

The Compare to an External Data Source dialog box allows you to open a tab-separated values file (with a .txt extension), a comma-separated values file (with a .csv extension), or a data file formatted for IBM® SPSS® Statistics (with a .sav extension). When the file is open, you can select a field in the data file to compare to the feature labels in a specific map key. You can then correct any discrepancies in the map file.

Fields in the data file. Choose the field whose values you want to compare to the feature labels. If the first row in the .txt or .csv file contains descriptive labels for each field, check Use first row as column labels. Otherwise, each field will be identified by its position in the data file (for example, "Column 1", "Column 2", and so on).

Key to compare. Choose the map key whose feature labels you want to compare to the data file field values.

Compare. Click when you are ready to compare the values.

Comparison Results. By default, the Comparison Results table lists only the unmatched field values in the data file. The application tries to find a related feature label, usually by checking for inserted or missing spaces. Click the dropdown list in the *Map Label* column to match the feature label in the map file to the displayed field value. If there is no corresponding feature label in your map file, choose *Leave Unmatched*. If you want to see all field values, even those that already match a feature label, clear Display only unmatched cases. You may do this if you want to override one or more matches.

You can use each feature only once to match it to a field value. If you want to match multiple features to a single field value, you can merge the features and then match the new, merged feature to the field value. For more information about merging features, see [Merge features](#).

Related information

- [Edit the feature labels](#)

Merge features

Merging features is useful for creating larger regions in a map. For example, if you were converting a map of states, you could merge the states (the features in this example) into larger North, South, East, and West regions.

Keys. Select the map key containing the feature labels that will help you identify the features you want to merge.

Features. Click the first feature that you want to merge. Ctrl-click the other features that you want to merge. Note that the features will also be selected in the map preview. You can click and Ctrl-click the features directly in the map preview in addition to selecting them from the list.

After selecting the features that you want to merge, click Merge to display the Name the Merged Feature dialog box, where you will be able to apply a label to the new feature. You may want to check Color the map preview after merging features to ensure that the results are what you expect.

After merging features, you may also want to move the label for the new feature. You can do that in the *Edit the feature labels* task. See the topic [Edit the feature labels](#) for more information.

- [Name the Merged Feature dialog box](#)

Related information

- [Step 3 - Edit the Map](#)
- [Smooth the map](#)
- [Edit the feature labels](#)
- [Move features](#)
- [Delete features](#)
- [Delete individual elements](#)
- [Set the projection](#)
- [Name the Merged Feature dialog box](#)

Name the Merged Feature dialog box

The Name the Merged Feature dialog box allows you to assign labels to the new merged feature.

The Labels table displays information for each key in the map file and allows you to assign a label for each key.

New Label. Enter a new label for the merged feature to assign to the specific map key.

Key. The map key to which you are assigning the new label.

Old Labels. The labels for the features that will be merged into the new feature.

Remove boundaries between touching polygons. Check this option to remove the boundaries from the features that were merged together. For example, if you merged states into geographic regions, this option would remove the boundaries around the individual states.

Related information

- [Merge features](#)
-

Move features

You can move features in the map. This can be useful when you want to put features together, like a mainland and outlying islands.

Keys. Select the map key containing the feature labels that will help you identify the features you want to move.

Features. Click the feature that you want to move. Note that the feature will be selected in the map preview. You can also click the feature directly in the map preview.

X/Y. These text boxes list the current center point of the feature on the map. The units are displayed in the map's coordinates. These may be local, Cartesian coordinates (for example, the State Plane Coordinate System) or geographic coordinates (where X is longitude and Y is latitude). Enter coordinates for the new position of the feature. You can also click and drag a feature on the map to move it. The text boxes will update with the new position.

Related information

- [Step 3 - Edit the Map](#)
 - [Smooth the map](#)
 - [Edit the feature labels](#)
 - [Merge features](#)
 - [Delete features](#)
 - [Delete individual elements](#)
 - [Set the projection](#)
-

Delete features

You can delete unwanted features from the map. This can be useful when you want to remove some clutter by deleting features that won't be of interest in the map visualization.

Keys. Select the map key containing the feature labels that will help you identify the features you want to delete.

Features. Click the feature that you want to delete. If you want to delete multiple features simultaneously, Ctrl-click the additional features. Note that the features will also be selected in the map preview. You can click and Ctrl-click the features directly in the map preview in addition to selecting them from the list.

Related information

- [Step 3 - Edit the Map](#)
 - [Smooth the map](#)
 - [Edit the feature labels](#)
 - [Merge features](#)
 - [Move features](#)
 - [Delete individual elements](#)
 - [Set the projection](#)
-

Delete individual elements

In addition to deleting whole features, you can delete some of the individual elements that compose the features, such as lakes and small islands. This option is not available for point maps.

Elements. Click the elements that you want to delete. If you want to delete multiple elements simultaneously, Ctrl-click the additional elements. Note that the elements will also be selected in the map preview. You can click and Ctrl-click the elements directly in the map preview in addition to selecting them from the list. Because the list of element names is not descriptive (each element is assigned a number within the feature), you should check the selection in the map preview to ensure you have selected the elements you want.

Related information

- [Step 3 - Edit the Map](#)
 - [Smooth the map](#)
 - [Edit the feature labels](#)
 - [Merge features](#)
 - [Move features](#)
 - [Delete features](#)
 - [Set the projection](#)
-

Set the projection

The map projection specifies the way in which the three-dimensional Earth is represented in two dimensions. All projections cause distortions. However, some projections are more suitable depending on whether you are viewing a global map or a more local one. Also some projections preserve the shape of the original features. Projections that preserve shape are conformal projections. This option is available only for maps with geographic coordinates (longitude and latitude).

Unlike other options in the Map Conversion Utility, the projection can be changed after a map visualization is created.

Projection. Select a map projection. If you are creating a global or hemispherical map, use the *Local*, *Mercator*, or *Winkel Tripel* projections. For smaller areas, use the *Local*, *Lambert Conformal Conic*, or *Transverse Mercator* projections. All projections use the WGS83 ellipsoid for the datum.

- The **Local** projection is always used when the map was created with a local coordinate system, such as the State Plane Coordinate System. These coordinate systems are defined by Cartesian coordinates rather than geographic coordinates (longitude and latitude). In the Local projection, horizontal and vertical lines are equally spaced in a Cartesian coordinate system. The Local projection is not conformal.
- The **Mercator** projection is a conformal projection for global maps. Horizontal and vertical lines are straight and always perpendicular to each other. Note that the Mercator projection extends to infinity as it approaches the North and South Poles, so it cannot be used if your map includes the North or South Pole. Distortion is greatest when the map approaches these limits.
- The **Winkel Tripel** projection is a non-conformal projection for global maps. Although it is not conformal, it provides a good balance between shape and size. Except for the Equator and Prime Meridian, all lines are curved. If your global map includes the North or South Pole, this is a good projection choice.
- As its name suggests, the **Lambert Conformal Conic** projection is a conformal projection and is used for maps of continental or smaller land masses that are longer East and West compared to North and South.
- The **Transverse Mercator** is another conformal projection for maps of continental or smaller land masses. Use this projection for land masses that are longer North and South compared to East and West.

Related information

- [Step 3 - Edit the Map](#)
 - [Smooth the map](#)
 - [Edit the feature labels](#)
 - [Merge features](#)
 - [Move features](#)
 - [Delete features](#)
 - [Delete individual elements](#)
-

Step 4 - Finish

At this point you can add a comment to describe the map file and also create a sample data file from the map keys.

Map keys. If there are multiple keys in the map file, select a map key whose feature labels you want to display in the preview. If you create a data file from the map, these labels will be used for the data values.

Comment. Enter a comment that describes the map or provides additional information that may be relevant to your users, such as the sources for the original shapefiles. The comment will appear in the Graphboard Template Chooser's Manage system.

Create a data set from the feature labels. Check this option if you want to create a data file from the displayed feature labels. When you click Browse..., you will be able to specify a location and file name. If you add a .txt extension, the file will be saved as a tab-separate values file. If you add a .csv extension, the file will be saved as a comma-separated values file. If you add a .sav extension, the file will be saved in IBM® SPSS® Statistics format. SAV is the default when no extension is specified.

Related information

- [Using the Map Conversion Utility](#)
 - [Step 1 - Choose Destination and Source Files](#)
 - [Step 2 - Choose Map Key](#)
 - [Step 3 - Edit the Map](#)
-

Distributing map files

In the first step of the Map Conversion Utility, you chose a location where to save the converted SMZ file. You may have also chosen to add the map to the Manage system for the Graphboard Template Chooser. If you chose to save to the Manage system, the map will be available to you in any IBM® SPSS® product that you run on the same computer.

To distribute the map to other users, you will need to send the SMZ to them. Those users can then use the Manage system to import the map. You can simply send the file whose location you specified in step 1. If you want to send a file that's in the Manage system, you first need to export it:

1. In the Template Chooser, click Manage...
2. Click the Map tab.
3. Select the map you want to distribute.
4. Click Export... and choose a location where you want to save the file.

Now you can send the physical map file to other users. Users will need to reverse this process and import the map into the Manage system.

Related information

- [Converting and Distributing Map Shapefiles](#)
 - [Key Concepts for Maps](#)
 - [Using the Map Conversion Utility](#)
-

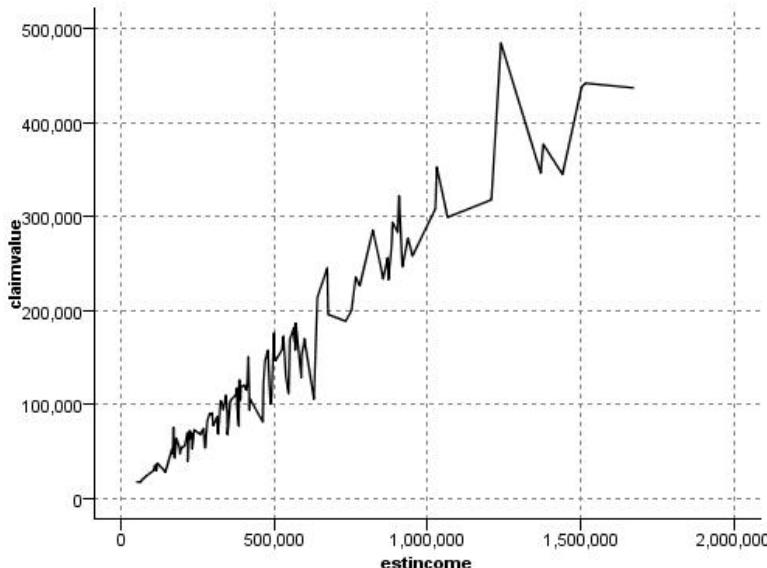
Plot Node

Plot nodes show the relationship between numeric fields. You can create a plot using points (also known as a scatterplot), or you can use lines. You can create three types of line plots by specifying an X Mode in the dialog box.

X Mode = Sort

Setting X Mode to Sort causes data to be sorted by values for the field plotted on the x axis. This produces a single line running from left to right on the graph. Using a nominal field as an overlay produces multiple lines of different hues running from left to right on the graph.

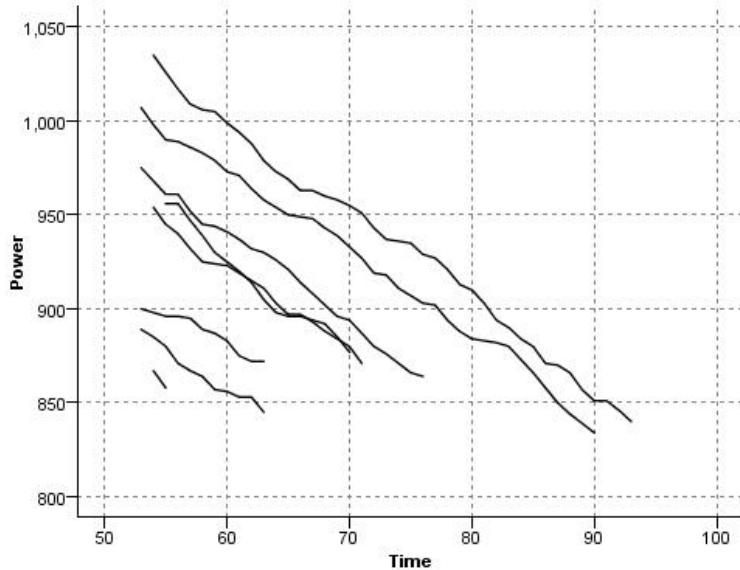
Figure 1. Line plot with X Mode set to Sort



X Mode = Overlay

Setting X Mode to Overlay creates multiple line plots on the same graph. Data are not sorted for an overlay plot; as long as the values on the x axis increase, data will be plotted on a single line. If the values decrease, a new line begins. For example, as x moves from 0 to 100, the y values will be plotted on a single line. When x falls below 100, a new line will be plotted in addition to the first one. The finished plot might have numerous plots useful for comparing several series of y values. This type of plot is useful for data with a periodic time component, such as electricity demand over successive 24-hour periods.

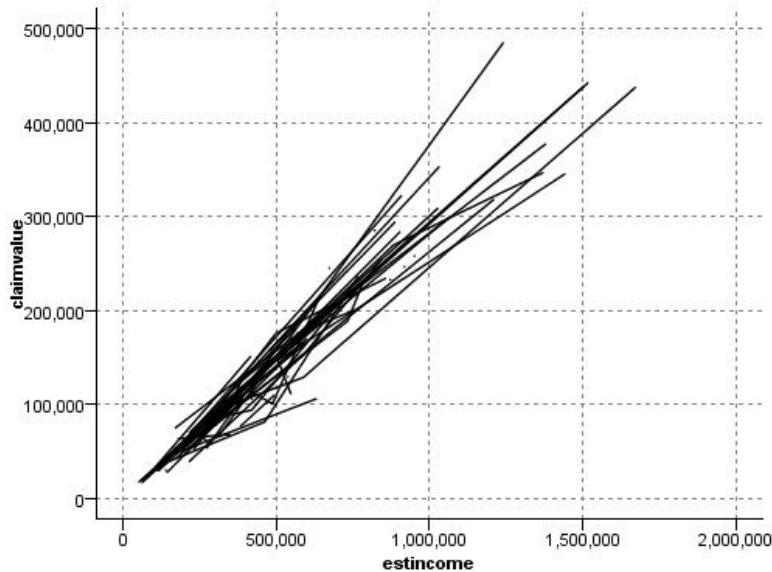
Figure 2. Line plot with X Mode set to Overlay



X Mode = As Read

Setting X Mode to As Read plots x and y values as they are read from the data source. This option is useful for data with a time series component where you are interested in trends or patterns that depend on the order of the data. You may need to sort the data before creating this type of plot. It may also be useful to compare two similar plots with X Mode set to Sort and As Read in order to determine how much of a pattern depends on the sorting.

Figure 3. Line plot shown earlier as Sort, executed again with X Mode set to As Read



You can also use the Graphboard node to produce scatterplots and line plots. However, you have more options to choose from in this node. See the topic [Available Built-in Graphboard Visualization Types](#) for more information.

- [Plot Node Tab](#)

- [Plot Options Tab](#)
- [Plot Appearance Tab](#)
- [Using a Plot Graph](#)

Related information

- [Plot Node Tab](#)
- [Plot Options Tab](#)
- [Plot Appearance Tab](#)
- [Using a Plot Graph](#)
- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Distribution Node](#)
- [Histogram Node](#)
- [Collection Node](#)
- [Multiplot Node](#)
- [Web Node](#)
- [Time Plot Node](#)
- [Evaluation node](#)
- [Working with Graph Output](#)
- [Exploring Graphs](#)
- [Editing Visualizations](#)
- [Adding Titles and Footnotes](#)
- [Using Graph Stylesheets](#)
- [Printing, saving, copying, and exporting graphs](#)

Plot Node Tab

Plots show values of a *Y* field against values of an *X* field. Often, these fields correspond to a dependent variable and an independent variable, respectively.

X field. From the list, select the field to display on the horizontal *x* axis.

Y field. From the list, select the field to display on the vertical *y* axis.

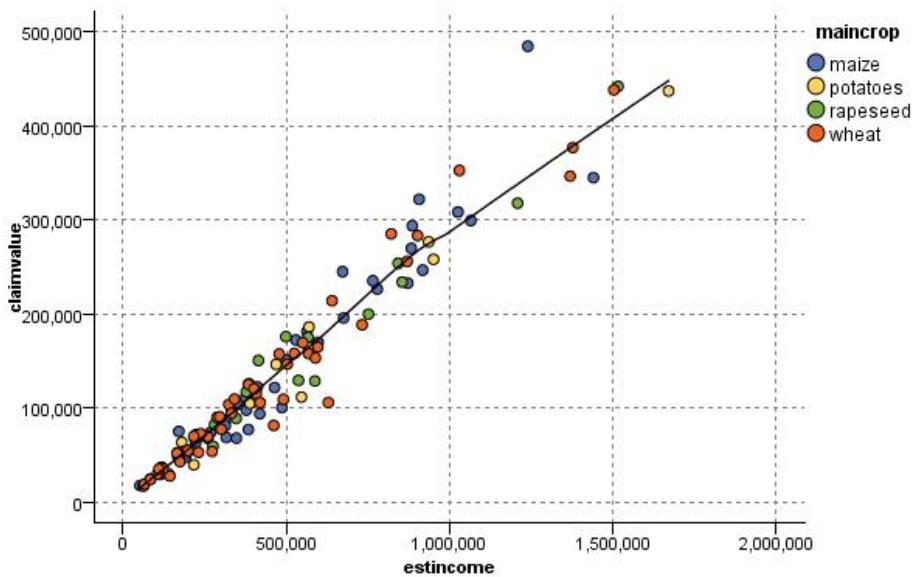
Z field. When you click the 3-D chart button, you can then choose a field from the list to display on the *z* axis.

Overlay. There are several ways to illustrate categories for data values. For example, you can use *maincrop* as a color overlay to indicate the *estincome* and *claimvalue* values for the main crop grown by claim applicants. See the topic [Aesthetics, Overlays, Panels, and Animation](#) for more information.

Overlay type. Specifies whether an overlay function or smoother is displayed. The smoother and overlay functions are always calculated as a function of *y*.

- **None.** No overlay is displayed.
- **Smoother.** Displays a smoothed fit line computed using locally weighted iterative robust least squares regression (LOESS). This method effectively computes a series of regressions, each focused on a small area within the plot. This produces a series of "local" regression lines that are then joined to create a smooth curve.

Figure 1. Plot with a LOESS smoother overlay



- **Function.** Select to specify a known function to compare to actual values. For example, to compare actual versus predicted values, you can plot the function $y = x$ as an overlay. Specify a function for $y =$ in the text box. The default function is $y = x$, but you can specify any sort of function, such as a quadratic function or an arbitrary expression, in terms of x .
- Note:* Overlay functions are not available for a panel or animation graph.

Once you have set options for a plot, you can run the plot directly from the dialog box by clicking Run. You may, however, want to use the Options tab for additional specifications, such as binning, X Mode, and style.

Related information

- [Plot Node](#)
 - [Plot Options Tab](#)
 - [Plot Appearance Tab](#)
 - [Using a Plot Graph](#)
-

Plot Options Tab

Style. Select either Point or Line for the plot style. Selecting Line activates the X Mode control. Selecting Point will use a plus symbol (+) as the default point shape. Once the graph is created, you can change the point shape and alter its size.

X Mode. For line plots, you must choose an X Mode to define the style of the line plot. Select Sort, Overlay, or As Read. See the topic [Plot Node](#) for more information. For Overlay or As Read, you should specify a maximum dataset size used to sample the first n records. Otherwise, the default 2,000 records will be used.

Automatic X range. Select to use the entire range of values in the data along this axis. Deselect to use an explicit subset of values based on your specified Min and Max values. Either enter values or use the arrows. Automatic ranges are selected by default to enable rapid graph building.

Automatic Y range. Select to use the entire range of values in the data along this axis. Deselect to use an explicit subset of values based on your specified Min and Max values. Either enter values or use the arrows. Automatic ranges are selected by default to enable rapid graph building.

Automatic Z range. Only when a 3-D graph is specified on the Plot tab. Select to use the entire range of values in the data along this axis. Deselect to use an explicit subset of values based on your specified Min and Max values. Either enter values or use the arrows. Automatic ranges are selected by default to enable rapid graph building.

Jitter. Also known as **agitation**, jitter is useful for point plots of a dataset in which many values are repeated. In order to see a clearer distribution of values, you can use jitter to distribute the points randomly around the actual value.

Note to users of earlier versions of IBM® SPSS® Modeler : The jitter value used in a plot uses a different metric in this release of IBM SPSS Modeler. In earlier versions, the value was an actual number, but it is now a proportion of the frame size. This means that agitation values in old streams are likely to be too large. For this release, any nonzero agitation values will be converted to the value 0.2.

Maximum number of records to plot. Specify a method for plotting large datasets. You can specify a maximum dataset size or use the default 2,000 records. Performance is enhanced for large datasets when you select the Bin or Sample options. Alternatively, you can choose to plot all

data points by selecting Use all data, but you should note that this may dramatically decrease the performance of the software.

Note: When X Mode is set to Overlay or As Read, these options are disabled and only the first *n* records are used.

- **Bin.** Select to enable binning when the dataset contains more than the specified number of records. Binning divides the graph into fine grids before actually plotting and counts the number of points that would appear in each of the grid cells. In the final graph, one point is plotted per cell at the bin centroid (average of all point locations in the bin). The size of the plotted symbols indicates the number of points in that region (unless you have used size as an overlay). Using the centroid and size to represent the number of points makes the binned plot a superior way to represent large datasets, because it prevents overplotting in dense regions (undifferentiated masses of color) and reduces symbol artifacts (artificial patterns of density). Symbol artifacts occur when certain symbols (particularly the plus symbol [+]) collide in a way that produces dense areas not present in the raw data.
- **Sample.** Select to randomly sample the data to the number of records entered in the text field. The default is 2,000.

Related information

- [Plot Node](#)
- [Plot Node Tab](#)
- [Plot Appearance Tab](#)
- [Using a Plot Graph](#)

Plot Appearance Tab

You can specify appearance options before graph creation.

Title. Enter the text to use for the graph's title.

Subtitle. Enter the text to use for the graph's subtitle.

Caption. Enter the text to use for the graph's caption.

X label. Either accept the automatically generated x-axis (horizontal) label or select Custom to specify a label.

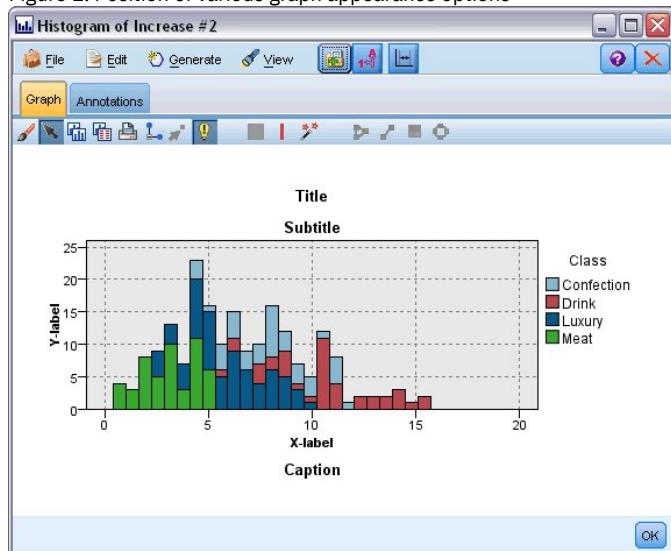
Y label. Either accept the automatically generated y-axis (vertical) label or select Custom to specify a label.

Z label. Available only for 3-D graphs, either accept the automatically generated z-axis label or select Custom to specify a custom label.

Display gridline. Selected by default, this option displays a gridline behind the plot or graph that enables you to more easily determine region and band cutoff points. Gridlines are always displayed in white unless the graph background is white; in this case, they are displayed in gray.

The following example shows where appearance options are placed on a graph. (Note that not all graphs use all these options.)

Figure 1. Position of various graph appearance options



Related information

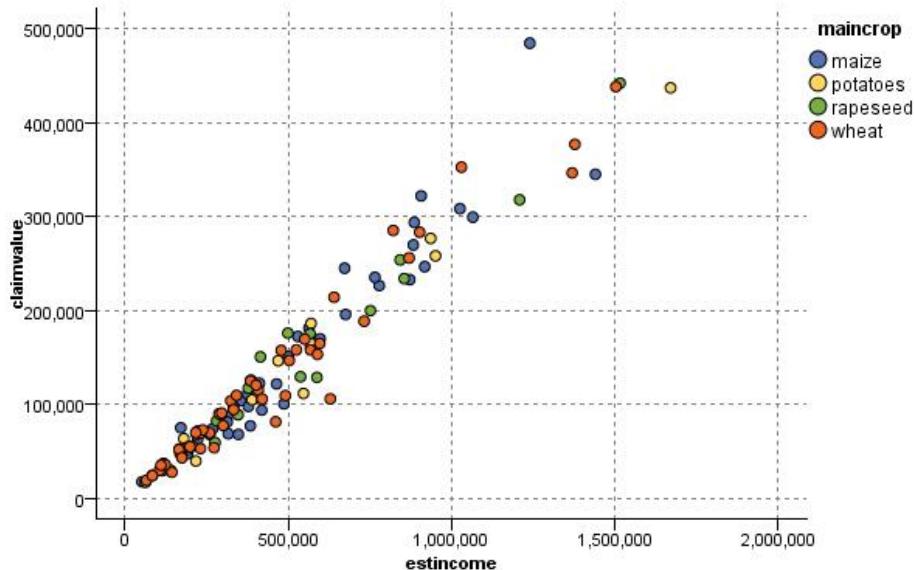
- [Plot Node](#)
- [Plot Node Tab](#)

- [Plot Options Tab](#)
- [Using a Plot Graph](#)

Using a Plot Graph

Plots and multiplots are essentially plots of X against Y . For example, if you are exploring potential fraud in agricultural grant applications, you might want to plot the income claimed on the application versus the income estimated by a neural net. Using an overlay, such as crop type, will illustrate whether there is a relationship between claims (value or number) and type of crop.

Figure 1. Plot of the relationship between estimated income and claim value with main crop type as an overlay



Since plots, multiplots, and evaluation charts are two-dimensional displays of Y against X , it is easy to interact with them by defining regions, marking elements, or even drawing bands. You can also generate nodes for the data represented by those regions, bands, or elements. See the topic [Exploring Graphs](#) for more information.

Related information

- [Plot Node](#)
- [Plot Node Tab](#)
- [Plot Options Tab](#)
- [Plot Appearance Tab](#)
- [Editing Visualizations](#)

Multiplot Node

A multiplot is a special type of plot that displays multiple Y fields over a single X field. The Y fields are plotted as colored lines and each is equivalent to a Plot node with Style set to Line and X Mode set to Sort. Multiplots are useful when you have time sequence data and want to explore the fluctuation of several variables over time.

- [Multiplot Plot Tab](#)
- [Multiplot Appearance Tab](#)
- [Using a Multiplot Graph](#)

Related information

- [Using a Plot Graph](#)
- [Multiplot Plot Tab](#)
- [Multiplot Appearance Tab](#)
- [Using a Multiplot Graph](#)

- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Plot Node](#)
- [Distribution Node](#)
- [Histogram Node](#)
- [Collection Node](#)
- [Web Node](#)
- [Time Plot Node](#)
- [Evaluation node](#)
- [Working with Graph Output](#)
- [Exploring Graphs](#)
- [Editing Visualizations](#)
- [Adding Titles and Footnotes](#)
- [Using Graph Stylesheets](#)
- [Printing, saving, copying, and exporting graphs](#)

Multiplot Plot Tab

A multiplot is a special type of plot that displays multiple Y fields over a single X field.

X field. From the list, select the field to display on the horizontal x axis.

Y fields. Select one or more fields from the list to display over the range of X field values. Use the Field Chooser button to select multiple fields. Click the delete button to remove fields from the list.

Overlay. There are several ways to illustrate categories for data values. For example, you might use an animation overlay to display multiple plots for each value in the data. This is useful for sets containing upwards of 10 categories. When used for sets with more than 15 categories, you may notice a decrease in performance. See the topic [Aesthetics, Overlays, Panels, and Animation](#) for more information.

Normalize. Select to scale all Y values to the range 0–1 for display on the graph. Normalizing helps you explore the relationship between lines that might otherwise be obscured due to differences in the range of values for each series and is recommended when plotting multiple lines on the same graph, or when comparing plots in side-by-side panels. (Normalizing is not necessary when all data values fall within a similar range.)

Figure 1. Standard multiplot showing power-plant fluctuation over time (note that without normalizing, the plot for Pressure is impossible to see)

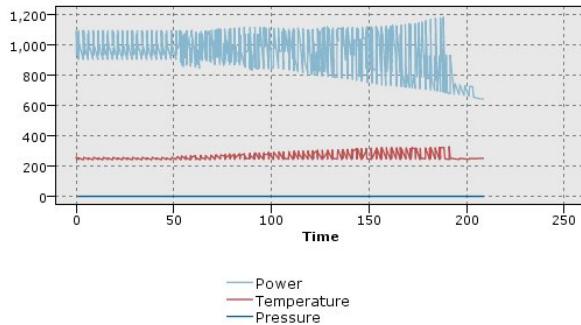
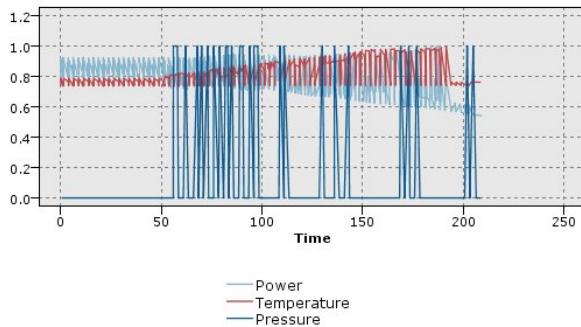


Figure 2. Normalized multiplot showing a plot for Pressure



Overlay function. Select to specify a known function to compare to actual values. For example, to compare actual versus predicted values, you can plot the function $y = x$ as an overlay. Specify a function for $y =$ in the text box. The default function is $y = x$, but you can specify any sort of function, such as a quadratic function or an arbitrary expression, in terms of x .

Note: Overlay functions are not available for a panel or animation graph.

When number of records greater than. Specify a method for plotting large datasets. You can specify a maximum dataset size or use the default 2,000 points. Performance is enhanced for large datasets when you select the Bin or Sample options. Alternatively, you can choose to plot all data points by selecting Use all data, but you should note that this may dramatically decrease the performance of the software.

Note: When X Mode is set to Overlay or As Read, these options are disabled and only the first n records are used.

- **Bin.** Select to enable binning when the dataset contains more than the specified number of records. Binning divides the graph into fine grids before actually plotting and counts the number of connections that would appear in each of the grid cells. In the final graph, one connection is used per cell at the bin centroid (average of all connection points in the bin).
- **Sample.** Select to randomly sample the data to the specified number of records.

Related information

- [Multiplot Node](#)
- [Multiplot Appearance Tab](#)
- [Using a Multiplot Graph](#)
- [Selecting Multiple Fields](#)

Multiplot Appearance Tab

You can specify appearance options before graph creation.

Title. Enter the text to use for the graph's title.

Subtitle. Enter the text to use for the graph's subtitle.

Caption. Enter the text to use for the graph's caption.

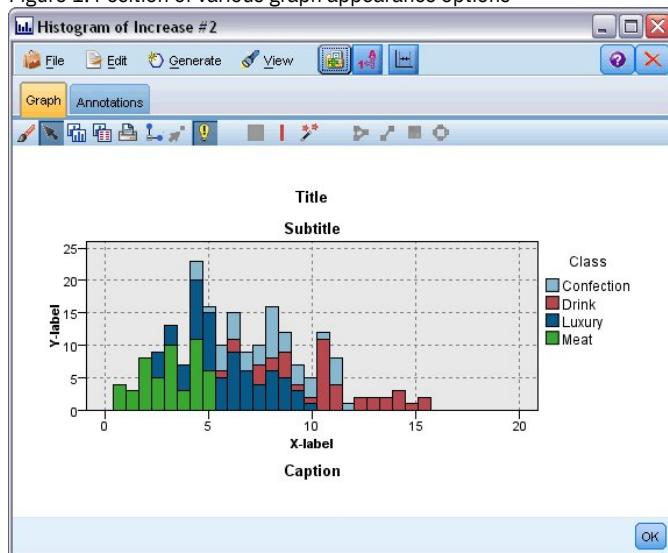
X label. Either accept the automatically generated x-axis (horizontal) label or select Custom to specify a label.

Y label. Either accept the automatically generated y-axis (vertical) label or select Custom to specify a label.

Display gridline. Selected by default, this option displays a gridline behind the plot or graph that enables you to more easily determine region and band cutoff points. Gridlines are always displayed in white unless the graph background is white; in this case, they are displayed in gray.

The following example shows where appearance options are placed on a graph. (Note that not all graphs use all these options.)

Figure 1. Position of various graph appearance options



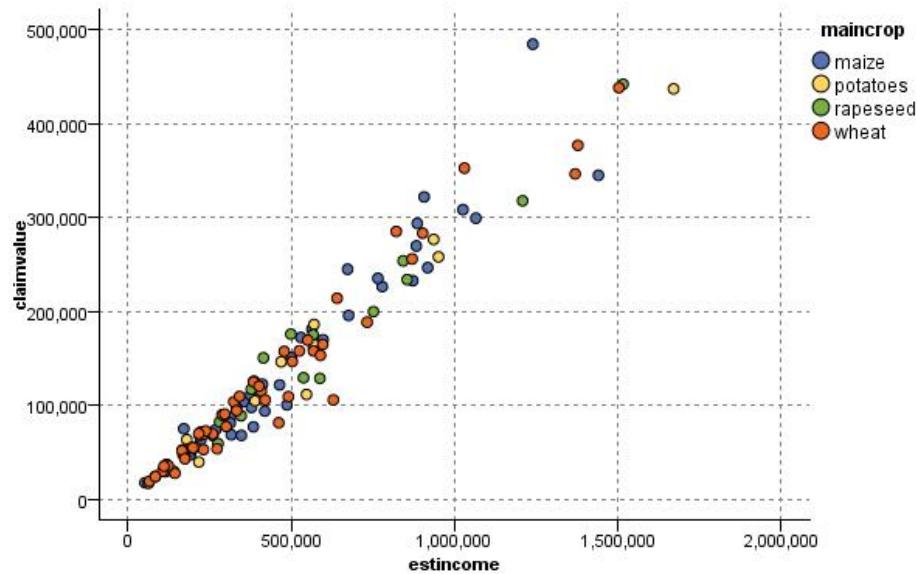
Related information

- [Multiplot Node](#)
- [Multiplot Plot Tab](#)
- [Using a Multiplot Graph](#)

Using a Multiplot Graph

Plots and multiplots are essentially plots of X against Y . For example, if you are exploring potential fraud in agricultural grant applications, you might want to plot the income claimed on the application versus the income estimated by a neural net. Using an overlay, such as crop type, will illustrate whether there is a relationship between claims (value or number) and type of crop.

Figure 1. Plot of the relationship between estimated income and claim value with main crop type as an overlay



Since plots, multiplots, and evaluation charts are two-dimensional displays of Y against X , it is easy to interact with them by defining regions, marking elements, or even drawing bands. You can also generate nodes for the data represented by those regions, bands, or elements. See the topic [Exploring Graphs](#) for more information.

Related information

- [Multiplot Node](#)
- [Multiplot Plot Tab](#)
- [Multiplot Appearance Tab](#)
- [Editing Visualizations](#)

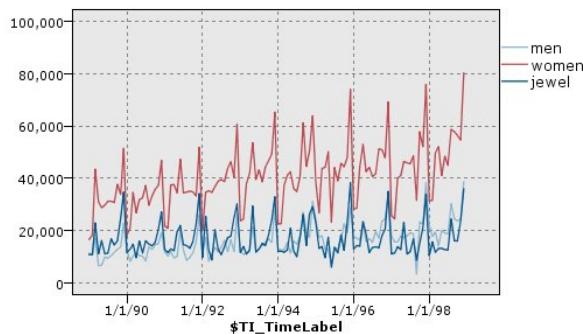
Time Plot Node

Time Plot nodes enable you to view one or more time series plotted over time. The series you plot must contain numeric values and are assumed to occur over a range of time in which the periods are uniform.

In SPSS® Modeler versions 17.1 and earlier, you usually use a Time Intervals node before a Time Plot node to create a *TimeLabel* field, which is used by default to label the x axis in the graphs.

For more information, see [Time Intervals Node \(deprecated\)](#).

Figure 1. Plotting sales of men's and women's clothing and jewelry over time



Creating Interventions and Events

You can create Event and Intervention fields from the time plot by generating a derive (flag or nominal) node from the context menus. For example, you could create an event field in the case of a rail strike, where the drive state is True if the event happened and False otherwise. For an Intervention field, for a price rise for example, you could use a derive count to identify the date of the rise, with 0 for the old price and 1 for the new price. See the topic [Derive node](#) for more information.

- [Time Plot Tab](#)
- [Time Plot Appearance Tab](#)
- [Using a Time Plot Graph](#)

Related information

- [Time Plot Tab](#)
- [Time Plot Appearance Tab](#)
- [Using a Time Plot Graph](#)
- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Plot Node](#)
- [Distribution Node](#)
- [Histogram Node](#)
- [Collection Node](#)
- [Multiplot Node](#)
- [Web Node](#)
- [Evaluation node](#)
- [Working with Graph Output](#)
- [Exploring Graphs](#)
- [Editing Visualizations](#)
- [Adding Titles and Footnotes](#)
- [Using Graph Stylesheets](#)
- [Printing, saving, copying, and exporting graphs](#)

Time Plot Tab

Plot. Provides a choice of how to plot time series data.

- **Selected series.** Plots values for selected time series. If you select this option when plotting confidence intervals, deselect the Normalize check box.
- **Selected Time Series models.** Used in conjunction with a Time Series model, this option plots all the related fields (actual and predicted values, as well as confidence intervals) for one or more selected time series. This option disables some other options in the dialog box. This is the preferred option if plotting confidence intervals.

Series. Select one or more fields with time series data you want to plot. The data must be numeric.

X axis label. Choose either the default label or a single field to use as the label for the x axis in plots. If you choose Default, the system uses the TimeLabel field created from a Time Intervals node upstream (for streams created in SPSS® Modeler versions 17.1 and earlier), or sequential integers if there is no upstream Time Intervals node.

For more information, see [Time Intervals Node \(deprecated\)](#).

Display series in separate panels. Specifies whether each series is displayed in a separate panel. Alternatively, if you do not choose to panel, all time series are plotted on the same graph, and smoothers will not be available. When plotting all time series on the same graph, each series will

be represented by a different color.

Normalize. Select to scale all Y values to the range 0–1 for display on the graph. Normalizing helps you explore the relationship between lines that might otherwise be obscured due to differences in the range of values for each series and is recommended when plotting multiple lines on the same graph, or when comparing plots in side-by-side panels. (Normalizing is not necessary when all data values fall within a similar range.)

Display. Select one or more elements to display in your plot. You can choose from lines, points, and (LOESS) smoothers. Smoothers are available only if you display the series in separate panels. By default, the line element is selected. Make sure you select at least one plot element before you run the graph node; otherwise, the system will return an error stating that you have selected nothing to plot.

Limit records. Select this option if you want to limit the number of records plotted. Specify the number of records, read from the beginning of your data file, that will be plotted in the Maximum number of records to plot option. By default this number is set to 2,000. If you want to plot the last *n* records in your data file, you can use a Sort node prior to this node to arrange the records in descending order by time.

Related information

- [Time Plot Node](#)
- [Time Plot Appearance Tab](#)
- [Using a Time Plot Graph](#)

Time Plot Appearance Tab

You can specify appearance options before graph creation.

Title. Enter the text to use for the graph's title.

Subtitle. Enter the text to use for the graph's subtitle.

Caption. Enter the text to use for the graph's caption.

X label. Either accept the automatically generated x-axis (horizontal) label or select Custom to specify a label.

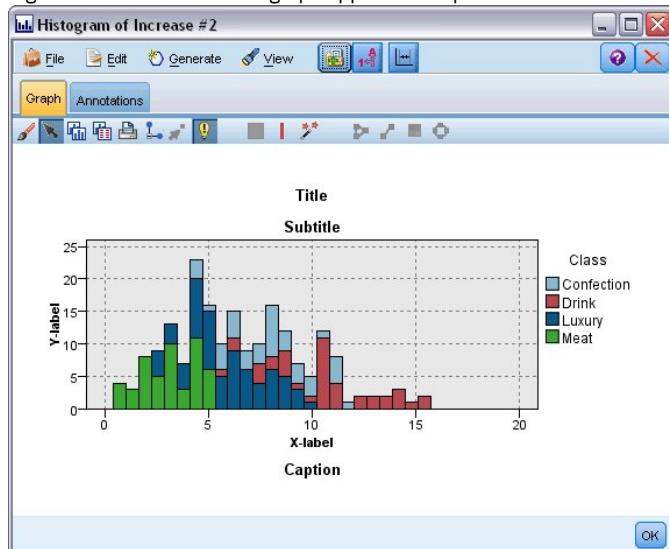
Y label. Either accept the automatically generated y-axis (vertical) label or select Custom to specify a label.

Display gridline. Selected by default, this option displays a gridline behind the plot or graph that enables you to more easily determine region and band cutoff points. Gridlines are always displayed in white unless the graph background is white; in this case, they are displayed in gray.

Layout. For time plots only, you can specify whether time values are plotted along a horizontal or vertical axis.

The following example shows where appearance options are placed on a graph. (Note that not all graphs use all these options.)

Figure 1. Position of various graph appearance options



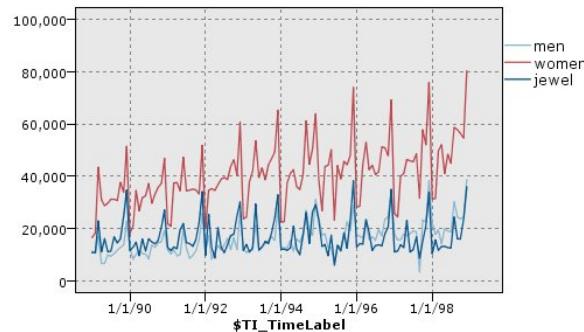
Related information

- [Time Plot Node](#)
- [Time Plot Tab](#)
- [Using a Time Plot Graph](#)

Using a Time Plot Graph

Once you have created a time plot graph, there are several options for adjusting the graph display and generating nodes for further analysis. See the topic [Exploring Graphs](#) for more information.

Figure 1. Plotting sales of men's and women's clothing and jewelry over time



Once you have created a time plot, defined bands, and examined the results, you can use options on the Generate menu and the context menu to create Select or Derive nodes. See the topic [Generating Nodes from Graphs](#) for more information.

Related information

- [Time Plot Node](#)
- [Time Plot Tab](#)
- [Time Plot Appearance Tab](#)
- [Editing Visualizations](#)

Distribution Node

A distribution graph or table shows the occurrence of symbolic (non-numeric) values, such as mortgage type or gender, in a dataset. A typical use of the Distribution node is to show imbalances in the data that can be rectified by using a Balance node before creating a model. You can automatically generate a Balance node using the Generate menu in the distribution graph or table window.

You can also use the Graphboard node to produce bar of counts graphs. However, you have more options to choose from in this node. See the topic [Available Built-in Graphboard Visualization Types](#) for more information.

Note: To show the occurrence of numeric values, you should use a Histogram node.

- [Distribution Plot Tab](#)
- [Distribution Appearance Tab](#)
- [Using a Distribution Node](#)

Related information

- [Distribution Plot Tab](#)
- [Distribution Appearance Tab](#)
- [Using a Distribution Node](#)
- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Plot Node](#)
- [Histogram Node](#)
- [Collection Node](#)
- [Multiplot Node](#)
- [Web Node](#)
- [Time Plot Node](#)
- [Evaluation node](#)
- [Working with Graph Output](#)
- [Exploring Graphs](#)
- [Editing Visualizations](#)
- [Adding Titles and Footnotes](#)

- [Using Graph Stylesheets](#)
 - [Printing, saving, copying, and exporting graphs](#)
-

Distribution Plot Tab

Plot. Select the type of distribution. Select Selected fields to show the distribution of the selected field. Select All flags (true values) to show the distribution of true values for flag fields in the dataset.

Field. Select a nominal or flag field for which to show the distribution of values. Only fields that have not been explicitly set as numeric appear on the list.

Overlay. Select a nominal or flag field to use as a color overlay, illustrating the distribution of its values within each value of the specified field. For example, you can use marketing campaign response (*pep*) as an overlay for number of children (*children*) to illustrate responsiveness by family size. See the topic [Aesthetics, Overlays, Panels, and Animation](#) for more information.

Normalize by color. Select to scale bars so that all bars take up the full width of the graph. The overlay values equal a proportion of each bar, making comparisons across categories easier.

Sort. Select the method used to display values in the distribution graph. Select Alphabetic to use alphabetical order or By count to list values in decreasing order of occurrence.

Proportional scale. Select to scale the distribution of values so that the value with the largest count fills the full width of the plot. All other bars are scaled against this value. Deselecting this option scales bars according to the total counts of each value.

Related information

- [Distribution Node](#)
 - [Distribution Appearance Tab](#)
 - [Using a Distribution Node](#)
-

Distribution Appearance Tab

You can specify appearance options before graph creation.

Title. Enter the text to use for the graph's title.

Subtitle. Enter the text to use for the graph's subtitle.

Caption. Enter the text to use for the graph's caption.

X label. Either accept the automatically generated x-axis (horizontal) label or select Custom to specify a label.

Y label. Either accept the automatically generated y-axis (vertical) label or select Custom to specify a label.

Display gridline. Selected by default, this option displays a gridline behind the plot or graph that enables you to more easily determine region and band cutoff points. Gridlines are always displayed in white unless the graph background is white; in this case, they are displayed in gray.

The following example shows where appearance options are placed on a graph. (Note that not all graphs use all these options.)

Figure 1. Position of various graph appearance options



Related information

- [Distribution Node](#)
- [Distribution Plot Tab](#)
- [Using a Distribution Node](#)

Using a Distribution Node

Distribution nodes are used to show the distribution of symbolic values in a dataset. They are frequently used before manipulation nodes to explore the data and correct any imbalances. For example, if instances of respondents without children occur much more frequently than other types of respondents, you might want to reduce these instances so that a more useful rule can be generated in later data mining operations. A Distribution node will help you to examine and make decisions about such imbalances.

The Distribution node is unusual in that it produces both a graph and a table to analyze your data.

Figure 1. Distribution graph showing the number of people with or without children who responded to a marketing campaign

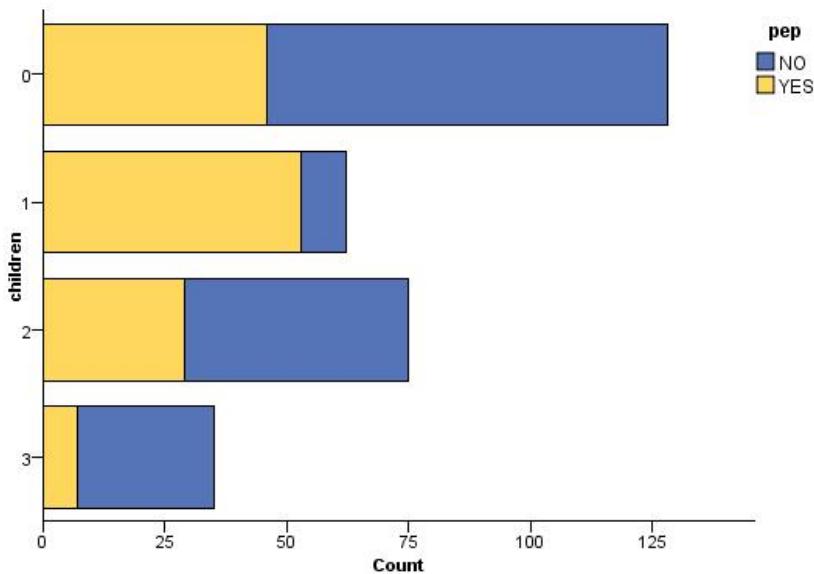
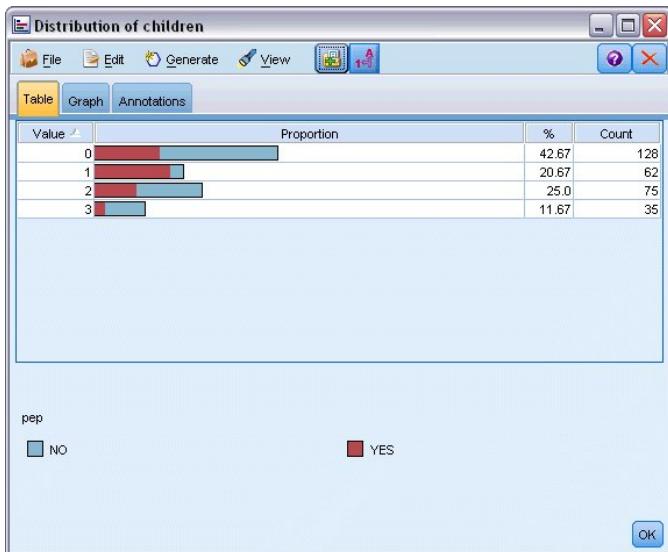


Figure 2. Distribution table showing the proportion of people with or without children who responded to a marketing campaign



Once you have created a distribution table and graph and examined the results, you can use options from the menus to group values, copy values, and generate a number of nodes for data preparation. In addition, you can copy or export the graph and table information for use in other applications, such as MS Word or MS PowerPoint. See the topic [Printing, saving, copying, and exporting graphs](#) for more information.

To Select and Copy Values from a Distribution Table

1. Click and hold the mouse button while dragging it over the rows to select a set of values. You can also use the Edit menu to Select All values.
 2. From the Edit menu, choose Copy Table or Copy Table (inc. field names).
 3. Paste to the clipboard or into the desired application.
- Note:* The bars do not get copied directly. Instead, the table values are copied. This means that overlaid values will not be displayed in the copied table.

To Group Values from a Distribution Table

1. Select values for grouping using the Ctrl+click method.
2. From the Edit menu, choose Group.

Note: When you group and ungroup values, the graph on the Graph tab is automatically redrawn to show the changes.

You can also:

- Ungroup values by selecting the group name in the distribution list and choosing Ungroup from the Edit menu.
- Edit groups by selecting the group name in the distribution list and choosing Edit group from the Edit menu. This opens a dialog box where values can be shifted to and from the group.

Generate Menu Options

You can use options on the Generate menu to select a subset of data, derive a flag field, regroup values, reclassify values, or balance the data from either a graph or table. These operations generate a data preparation node and place it on the stream canvas. To use the generated node, connect it to an existing stream. See the topic [Generating Nodes from Graphs](#) for more information.

Related information

- [Distribution Node](#)
 - [Distribution Plot Tab](#)
 - [Distribution Appearance Tab](#)
 - [Editing Visualizations](#)
-

Histogram Node

Histogram nodes show the occurrence of values for numeric fields. They are often used to explore the data before manipulations and model building. Similar to the Distribution node, Histogram nodes are frequently used to reveal imbalances in the data. While you can also use the Graphboard node to produce a histogram, you have more options to choose from in this node. See the topic [Available Built-in Graphboard Visualization Types](#) for more information.

Note: To show the occurrence of values for symbolic fields, you should use a Distribution node.

- [Histogram Plot Tab](#)
- [Histogram Options Tab](#)
- [Histogram Appearance Tab](#)
- [Using Histograms](#)

Related information

- [Histogram Plot Tab](#)
- [Histogram Options Tab](#)
- [Histogram Appearance Tab](#)
- [Using Histograms](#)
- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Plot Node](#)
- [Distribution Node](#)
- [Collection Node](#)
- [Multiplot Node](#)
- [Web Node](#)
- [Time Plot Node](#)
- [Evaluation node](#)
- [Working with Graph Output](#)
- [Exploring Graphs](#)
- [Editing Visualizations](#)
- [Adding Titles and Footnotes](#)
- [Using Graph Stylesheets](#)
- [Printing, saving, copying, and exporting graphs](#)

Histogram Plot Tab

Field. Select a numeric field for which to show the distribution of values. Only fields that have not been explicitly defined as symbolic (categorical) will be listed.

Overlay. Select a symbolic field to show categories of values for the specified field. Selecting an overlay field converts the histogram to a stacked chart with colors used to represent different categories of the overlay field. Using the Histogram node, there are three types of overlays: color, panel, and animation. See the topic [Aesthetics, Overlays, Panels, and Animation](#) for more information.

Related information

- [Histogram Node](#)
- [Histogram Options Tab](#)
- [Histogram Appearance Tab](#)
- [Using Histograms](#)

Histogram Options Tab

Automatic X range. Select to use the entire range of values in the data along this axis. Deselect to use an explicit subset of values based on your specified Min and Max values. Either enter values or use the arrows. Automatic ranges are selected by default to enable rapid graph building.

Bins. Select either By number or By width.

- Select By number to display a fixed number of bars whose width depends on the range and the number of bins specified. Indicate the number of bins to be used in the graph in the No. of bins option. Use the arrows to adjust the number.
- Select By width to create a graph with bars of a fixed width. The number of bins depends on the specified width and the range of values. Indicate the width of the bars in the Bin width option.

Normalize by color. Select to adjust all bars to the same height, displaying overlaid values as a percentage of the total cases in each bar.

Show normal curve. Select to add a normal curve to the graph showing the mean and variance of the data.

Separate bands for each color. Select to display each overlaid value as a separate band on the graph.

Related information

- [Histogram Node](#)
- [Histogram Plot Tab](#)
- [Histogram Appearance Tab](#)
- [Using Histograms](#)

Histogram Appearance Tab

You can specify appearance options before graph creation.

Title. Enter the text to use for the graph's title.

Subtitle. Enter the text to use for the graph's subtitle.

Caption. Enter the text to use for the graph's caption.

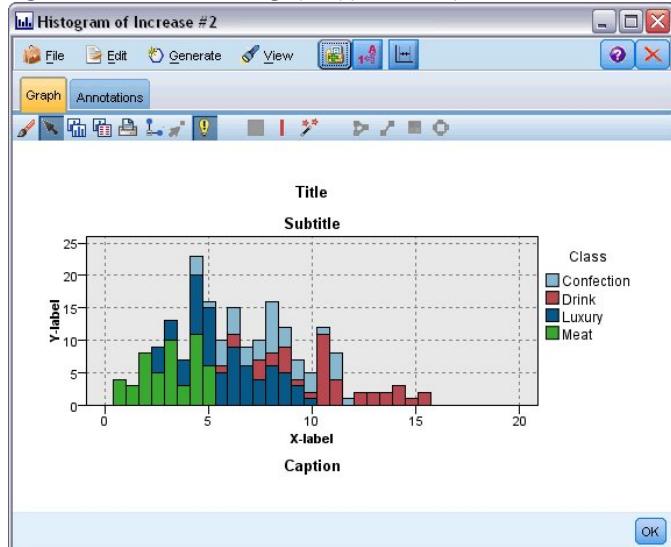
X label. Either accept the automatically generated x-axis (horizontal) label or select Custom to specify a label.

Y label. Either accept the automatically generated y-axis (vertical) label or select Custom to specify a label.

Display gridline. Selected by default, this option displays a gridline behind the plot or graph that enables you to more easily determine region and band cutoff points. Gridlines are always displayed in white unless the graph background is white; in this case, they are displayed in gray.

The following example shows where appearance options are placed on a graph. (Note that not all graphs use all these options.)

Figure 1. Position of various graph appearance options



Related information

- [Histogram Node](#)
- [Histogram Plot Tab](#)
- [Histogram Options Tab](#)
- [Using Histograms](#)

Using Histograms

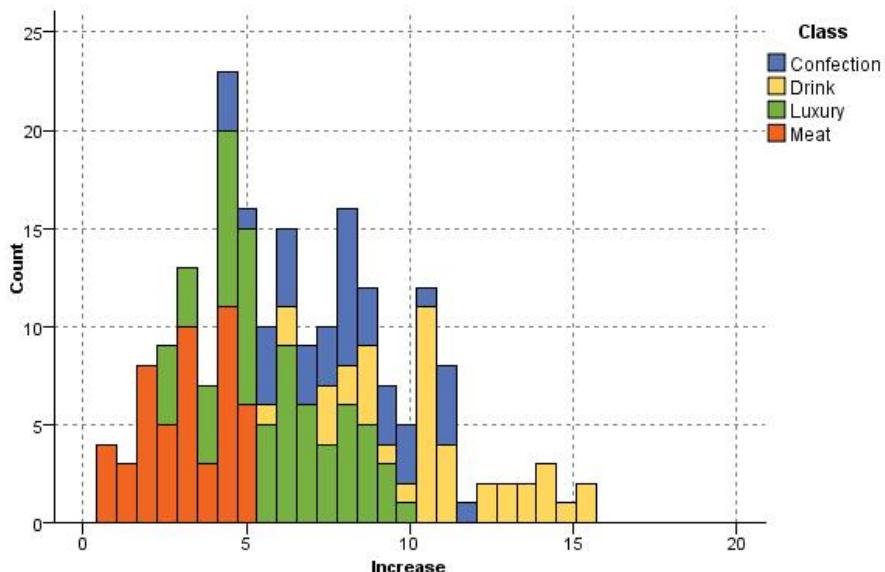
Histograms show the distribution of values in a numeric field whose values range along the x axis. Histograms operate similarly to collections graphs. Collections show the distribution of values for one numeric field *relative to the values of another*, rather than the occurrence of values for a single field.

Once you have created a graph, you can examine the results and define bands to split values along the x axis or define regions. You can also mark elements within the graph. See the topic [Exploring Graphs](#) for more information.

You can use options on the Generate menu to create Balance, Select, or Derive nodes using the data in the graph or more specifically within bands, regions, or marked elements. This type of graph is frequently used before manipulation nodes to explore the data and correct any imbalances by generating a Balance node from the graph to use in the stream. You can also generate a Derive Flag node to add a field showing

which band each record falls into or a Select node to select all records within a particular set or range of values. Such operations help you to focus on a particular subset of data for further exploration. See the topic [Generating Nodes from Graphs](#) for more information.

Figure 1. Histogram showing the distribution of increased purchases by category due to promotion



Related information

- [Histogram Node](#)
- [Histogram Plot Tab](#)
- [Histogram Options Tab](#)
- [Histogram Appearance Tab](#)
- [Editing Visualizations](#)
- [Select Node](#)
- [Balance Node](#)
- [Derive node](#)

Collection Node

Collections are similar to histograms except that collections show the distribution of values for one numeric field relative to the values of another, rather than the occurrence of values for a single field. A collection is useful for illustrating a variable or field whose values change over time. Using 3-D graphing, you can also include a symbolic axis displaying distributions by category. Two dimensional Collections are shown as stacked bar charts, with overlays where used. See the topic [Aesthetics, Overlays, Panels, and Animation](#) for more information.

- [Collection Plot Tab](#)
- [Collection Options Tab](#)
- [Collection Appearance Tab](#)
- [Using a Collection Graph](#)

Related information

- [Collection Plot Tab](#)
- [Collection Options Tab](#)
- [Collection Appearance Tab](#)
- [Using a Collection Graph](#)
- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Plot Node](#)
- [Distribution Node](#)
- [Histogram Node](#)
- [Multiplot Node](#)
- [Web Node](#)

- [Time Plot Node](#)
 - [Evaluation node](#)
 - [Working with Graph Output](#)
 - [Exploring Graphs](#)
 - [Editing Visualizations](#)
 - [Adding Titles and Footnotes](#)
 - [Using Graph Stylesheets](#)
 - [Printing, saving, copying, and exporting graphs](#)
-

Collection Plot Tab

Collect. Select a field whose values will be collected and displayed over the range of values for the field specified in Over. Only fields that have not been defined as symbolic are listed.

Over. Select a field whose values will be used to display the field specified in Collect.

By. Enabled when creating a 3-D graph, this option enables you to select a nominal or flag field used to display the collection field by categories.

Operation. Select what each bar in the collection graph represents. Options include Sum, Mean, Max, Min, and Standard Deviation.

Overlay. Select a symbolic field to show categories of values for the selected field. Selecting an overlay field converts the collection and creates multiple bars of varying colors for each category. This node has three types of overlays: color, panel, and animation. See the topic [Aesthetics, Overlays, Panels, and Animation](#) for more information.

Related information

- [Collection Node](#)
 - [Collection Options Tab](#)
 - [Collection Appearance Tab](#)
 - [Using a Collection Graph](#)
-

Collection Options Tab

Automatic X range. Select to use the entire range of values in the data along this axis. Deselect to use an explicit subset of values based on your specified Min and Max values. Either enter values or use the arrows. Automatic ranges are selected by default to enable rapid graph building.

Bins. Select either By number or By width.

- Select By number to display a fixed number of bars whose width depends on the range and the number of bins specified. Indicate the number of bins to be used in the graph in the No. of bins option. Use the arrows to adjust the number.
- Select By width to create a graph with bars of a fixed width. The number of bins depends on the specified width and the range of values. Indicate the width of the bars in the Bin width option.

Related information

- [Collection Node](#)
 - [Collection Plot Tab](#)
 - [Collection Appearance Tab](#)
 - [Using a Collection Graph](#)
-

Collection Appearance Tab

You can specify appearance options before graph creation.

Title. Enter the text to use for the graph's title.

Subtitle. Enter the text to use for the graph's subtitle.

Caption. Enter the text to use for the graph's caption.

Over label. Either accept the automatically generated label, or select Custom to specify a label.

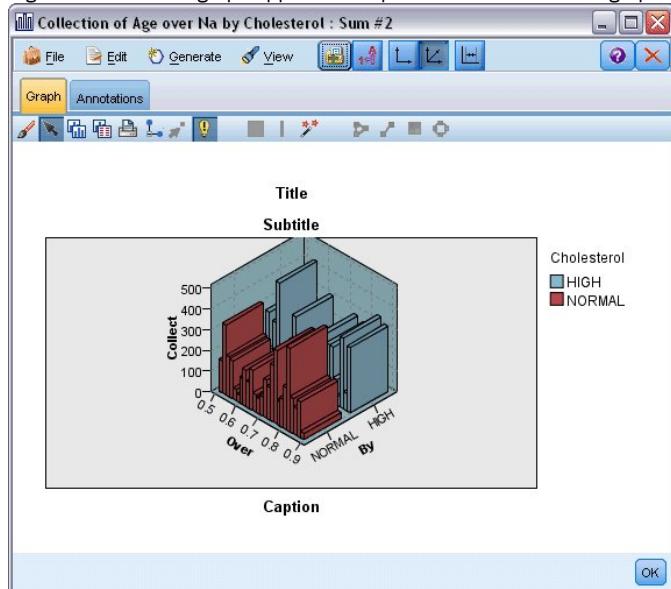
Collect label. Either accept the automatically generated label, or select Custom to specify a label.

By label. Either accept the automatically generated label, or select Custom to specify a label.

Display gridline. Selected by default, this option displays a gridline behind the plot or graph that enables you to more easily determine region and band cutoff points. Gridlines are always displayed in white unless the graph background is white; in this case, they are displayed in gray.

The following example shows where the appearance options are placed on a 3-D version of the graph.

Figure 1. Position of graph appearance options on 3-D Collection graph.



Related information

- [Collection Node](#)
- [Collection Plot Tab](#)
- [Collection Options Tab](#)
- [Using a Collection Graph](#)

Using a Collection Graph

Collections show the distribution of values for one numeric field *relative to the values of another*, rather than the occurrence of values for a single field. Histograms operate similarly to collections graphs. Histograms show the distribution of values in a numeric field whose values range along the x axis.

Once you have created a graph, you can examine the results and define bands to split values along the x axis or define regions. You can also mark elements within the graph. See the topic [Exploring Graphs](#) for more information.

You can use options on the Generate menu to create Balance, Select, or Derive nodes using the data in the graph or more specifically within bands, regions, or marked elements. This type of graph is frequently used before manipulation nodes to explore the data and correct any imbalances by generating a Balance node from the graph to use in the stream. You can also generate a Derive Flag node to add a field showing which band each record falls into or a Select node to select all records within a particular set or range of values. Such operations help you to focus on a particular subset of data for further exploration. See the topic [Generating Nodes from Graphs](#) for more information.

Figure 1. 3-D collection graph showing sum of Na_to_K over Age for both high and normal cholesterol levels

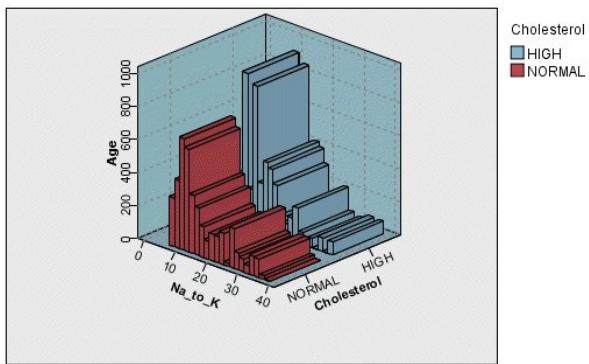
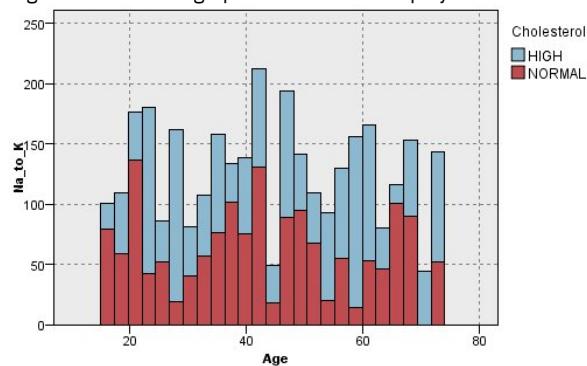


Figure 2. Collection graph without z axis displayed but with Cholesterol as color overlay



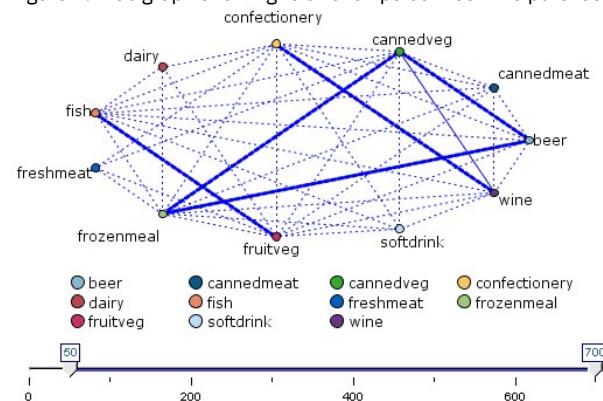
Related information

- [Collection Node](#)
- [Collection Plot Tab](#)
- [Collection Options Tab](#)
- [Collection Appearance Tab](#)
- [Editing Visualizations](#)

Web Node

Web nodes show the strength of relationships between values of two or more symbolic fields. The graph displays connections using varying types of lines to indicate connection strength. You can use a Web node, for example, to explore the relationship between the purchase of various items at an e-commerce site or a traditional retail outlet.

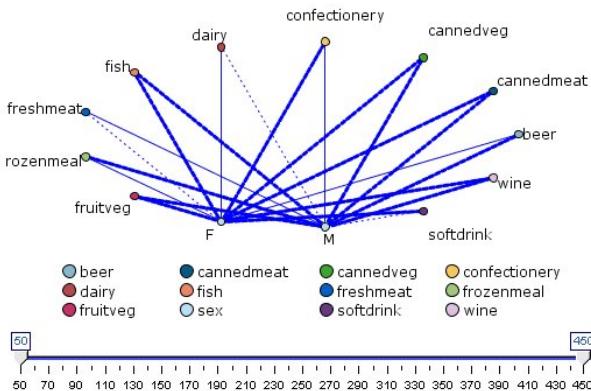
Figure 1. Web graph showing relationships between the purchase of grocery items



Directed Webs

Directed Web nodes are similar to Web nodes in that they show the strength of relationships between symbolic fields. However, directed web graphs show connections only from one or more From fields to a single To field. The connections are unidirectional in the sense that they are one-way connections.

Figure 2. Directed web graph showing the relationship between the purchase of grocery items and gender



Like Web nodes, the graph displays connections using varying types of lines to indicate connection strength. You can use a Directed Web node, for example, to explore the relationship between gender and a proclivity for certain purchase items.

- [Web Plot Tab](#)
- [Web Options Tab](#)
- [Web Appearance Tab](#)
- [Using a Web Graph](#)

Related information

- [Web Plot Tab](#)
 - [Web Options Tab](#)
 - [Web Appearance Tab](#)
 - [Using a Web Graph](#)
 - [Adjusting Web Thresholds](#)
 - [Creating a web summary](#)
 - [Common Graph Nodes Features](#)
 - [Graphboard node](#)
 - [Plot Node](#)
 - [Distribution Node](#)
 - [Histogram Node](#)
 - [Collection Node](#)
 - [Multiplot Node](#)
 - [Time Plot Node](#)
 - [Evaluation node](#)
 - [Working with Graph Output](#)
 - [Exploring Graphs](#)
 - [Editing Visualizations](#)
 - [Adding Titles and Footnotes](#)
 - [Using Graph Stylesheets](#)
 - [Printing, saving, copying, and exporting graphs](#)
-

Web Plot Tab

Web. Select to create a web graph illustrating the strength of relationships between all specified fields.

Directed web. Select to create a directional web graph illustrating the strength of relationships between multiple fields and the values of one field, such as gender or religion. When this option is selected, a To Field is activated and the Fields control below is renamed From Fields for additional clarity.

To Field (directed web only). Select a flag or nominal field used for a directed web. Only fields that have not been explicitly set as numeric are listed.

Fields/From Fields. Select fields to create a web graph. Only fields that have not been explicitly set as numeric are listed. Use the Field Chooser button to select multiple fields or select fields by type.

Note: For a directed web, this control is used to select From fields.

Show true flags only. Select to display only true flags for a flag field. This option simplifies the web display and is often used for data where the occurrence of positive values is of special importance.

Line values are. Select a threshold type from the drop-down list.

- Absolute sets thresholds based on the number of records having each pair of values.
- Overall percentages shows the absolute number of cases represented by the link as a proportion of all of the occurrences of each pair of values represented in the web graph.
- Percentages of smaller field/value and Percentages of larger field/value indicate which field/value to use for evaluating percentages. For example, suppose 100 records have the value *drugY* for the field *Drug* and only 10 have the value *LOW* for the field *BP*. If seven records have both values *drugY* and *LOW*, this percentage is either 70% or 7%, depending on which field you are referencing, smaller (*BP*) or larger (*Drug*).

Note: For directed web graphs, the third and fourth options above are not available. Instead, you can select Percentage of "To" field/value and Percentage of "From" field/value.

Strong links are heavier. Selected by default, this is the standard way of viewing links between fields.

Weak links are heavier. Select to reverse the meaning of links displayed in bold lines. This option is frequently used for fraud detection or examination of outliers.

Related information

- [Web Node](#)
- [Web Options Tab](#)
- [Web Appearance Tab](#)
- [Using a Web Graph](#)
- [Adjusting Web Thresholds](#)
- [Creating a web summary](#)

Web Options Tab

The Options tab for Web nodes contains a number of additional options to customize the output graph.

Number of Links. The following options are used to control the number of links displayed in the output graph. Some of these options, such as Weak links above and Strong links above, are also available in the output graph window. You can also use a slider control in the final graph to adjust the number of links displayed.

- Maximum number of links to display. Specify a number indicating the maximum number of links to show on the output graph. Use the arrows to adjust the value.
- Show only links above. Specify a number indicating the minimum value for which to show a connection in the web. Use the arrows to adjust the value.
- Show all links. Specify to display all links regardless of minimum or maximum values. Selecting this option may increase processing time if there are a large number of fields.

Discard if very few records. Select to ignore connections that are supported by too few records. Set the threshold for this option by entering a number in Min. records/line.

Discard if very many records. Select to ignore strongly supported connections. Enter a number in Max. records/line.

Weak links below. Specify a number indicating the threshold for weak connections (dotted lines) and regular connections (normal lines). All connections below this value are considered weak.

Strong links above. Specify a threshold for strong connections (heavy lines) and regular connections (normal lines). All connections above this value are considered strong.

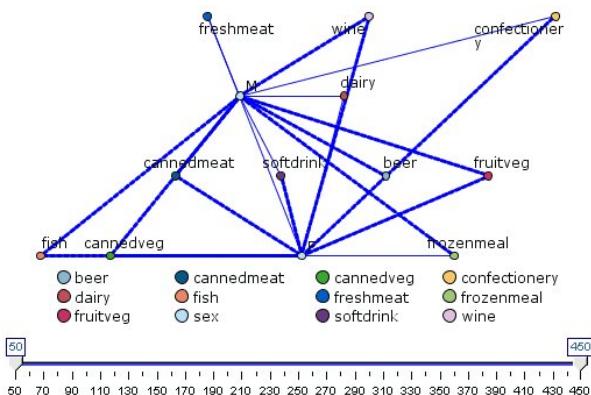
Link Size. Specify options for controlling the size of links:

- Link size varies continuously. Select to display a range of link sizes reflecting the variation in connection strengths based on actual data values.
- Link size shows strong/normal/weak categories. Select to display three strengths of connections--strong, normal, and weak. The cutoff points for these categories can be specified above as well as in the final graph.

Web Display. Select a type of web display:

- Circle layout. Select to use the standard web display.
- Network layout. Select to use an algorithm to group together the strongest links. This is intended to highlight strong links using spatial differentiation as well as weighted lines.
- Directed Layout. Select to create a directed web display that uses the To Field selection from the Plot tab as the focus for the direction.
- Grid Layout. Select to create a web display that is laid out in a regularly spaced grid pattern.

Figure 1. Web graph showing strong connections from frozenmeal and cannedveg to other grocery items



Note: When filtering the displayed links (using either the slider in the web graph or the Show only links above control on the Web node's Options tab), you may end up in a situation where all the links that remain to be displayed are of a single value (in other words, they're either all weak links, all medium links, or all strong links, as defined by the Weak links below and Strong links above controls on the Web node's Options tab). If this happens, all the links are all displayed with the medium width line in the web graph output.

Web Appearance Tab

You can specify appearance options before graph creation.

Title. Enter the text to use for the graph's title.

Subtitle. Enter the text to use for the graph's subtitle.

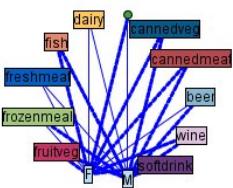
Caption. Enter the text to use for the graph's caption.

Show legend. You can specify whether the legend is displayed. For plots with a large number of fields, hiding the legend may improve the appearance of the plot.

Use labels as nodes. You can include the label text within each node rather than displaying adjacent labels. For plots with a small number of fields, this may result in a more readable chart.

Figure 1. Web graph showing labels as nodes

Relationship between gender and grocery purchases



Related information

- [Web Node](#)
- [Web Plot Tab](#)
- [Web Options Tab](#)
- [Using a Web Graph](#)
- [Adjusting Web Thresholds](#)
- [Creating a web summary](#).

Using a Web Graph

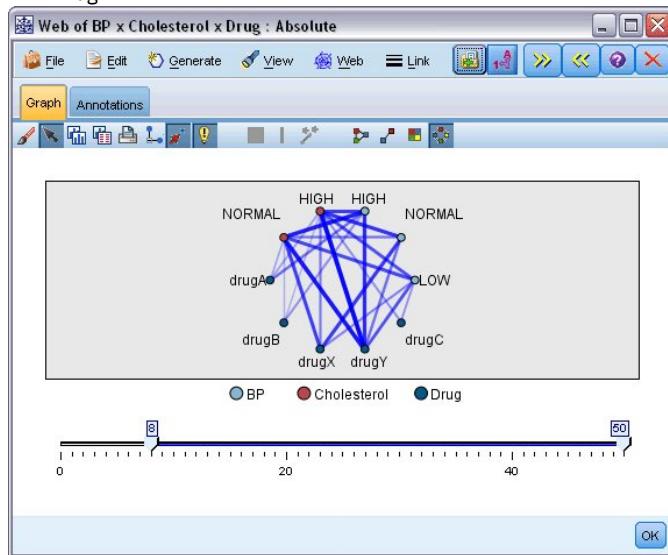
Web nodes are used to show the strength of relationships between values of two or more symbolic fields. Connections are displayed in a graph with varying types of lines to indicate connections of increasing strength. You can use a Web node, for example, to explore the relationship between cholesterol levels, blood pressure, and the drug that was effective in treating the patient's illness.

- Strong connections are shown with a heavy line. This indicates that the two values are strongly related and should be further explored.

- Medium connections are shown with a line of normal weight.
- Weak connections are shown with a dotted line.
- If no line is shown between two values, this means either that the two values never occur in the same record or that this combination occurs in a number of records below the threshold specified in the Web node dialog box.

Once you have created a Web node, there are several options for adjusting the graph display and generating nodes for further analysis.

Figure 1. Web graph indicating a number of strong relationships, such as normal blood pressure with DrugX and high cholesterol with DrugY



For both Web nodes and Directed Web nodes, you can:

- Change the layout of the web display.
- Hide points to simplify the display.
- Change the thresholds controlling line styles.
- Highlight lines between values to indicate a "selected" relationship.
- Generate a Select node for one or more "selected" records or a Derive Flag node associated with one or more relationships in the web.

To adjust points

- **Move** points by clicking the mouse on a point and dragging it to the new location. The web will be redrawn to reflect the new location.
- **Hide** points by right-clicking on a point in the web and choosing Hide or Hide and Replan from the context menu. Hide simply hides the selected point and any lines associated with it. Hide and Replan redraws the web, adjusting for any changes you have made. Any manual moves are undone.
- **Show** all hidden points by choosing Reveal All or Reveal All and Replan from the Web menu in the graph window. Selecting Reveal All and Replan redraws the web, adjusting to include all previously hidden points and their connections.

To select, or "highlight," lines

Selected lines are highlighted in red.

1. To select a single line, left-click the line.
2. To select multiple lines, do one of the following:
 - Using the cursor, draw a circle around the points whose lines you want to select.
 - Hold down the Ctrl key and left-click the individual lines you want to select.

You can deselect all selected lines by clicking the graph background, or by choosing Clear Selection from the Web menu in the graph window.

To view the web using a different layout

From the Web menu, choose Circle Layout, Network Layout, Directed Layout, or Grid Layout to change the layout of the graph.

To turn the links slider on or off

From the View menu, choose Links Slider.

To select or flag records for a single relationship

1. Right-click on the line representing the relationship of interest.
2. From the context menu, choose Generate Select Node For Link or Generate Derive Node For Link.

A Select node or Derive node is automatically added to the stream canvas with the appropriate options and conditions specified:

- The Select node selects all records in the given relationship.
- The Derive node generates a flag indicating whether the selected relationship holds true for records in the entire dataset. The flag field is named by joining the two values in the relationship with an underscore, such as `LOW_drugC` or `drugC_LOW`.

To select or flag records for a group of relationships

1. Select the line(s) in the web display representing relationships of interest.
 2. From the Generate menu in the graph window, choose Select Node ("And"), Select Node ("Or"), Derive Node ("And"), or Derive Node ("Or").
- The "Or" nodes give the disjunction of conditions. This means that the node will apply to records for which any of the selected relationships hold.
 - The "And" nodes give the conjunction of conditions. This means that the node will apply only to records for which all selected relationships hold. An error occurs if any of the selected relationships are mutually exclusive.

After you have completed your selection, a Select node or Derive node is automatically added to the stream canvas with the appropriate options and conditions specified.

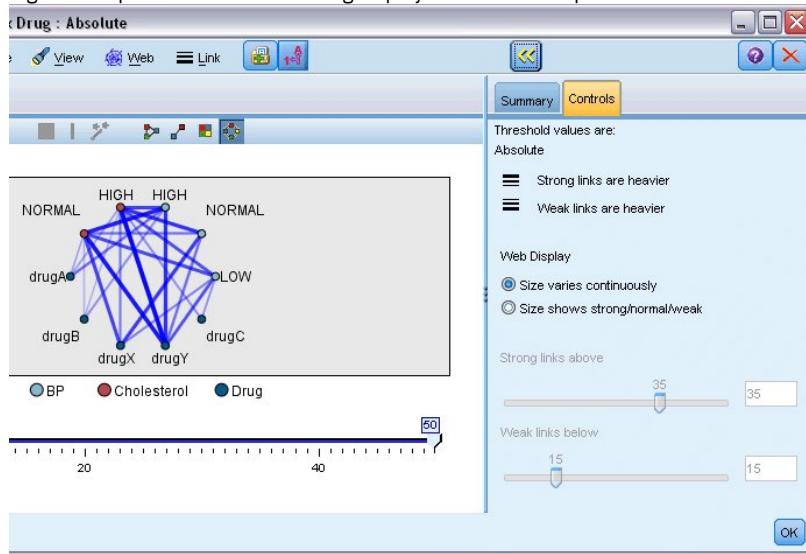
Note: When filtering the displayed links (using either the slider in the web graph or the Show only links above control on the Web node's Options tab), you may end up in a situation where all the links that remain to be displayed are of a single value (in other words, they're either all weak links, all medium links, or all strong links, as defined by the Weak links below and Strong links above controls on the Web node's Options tab). If this happens, all the links are all displayed with the medium width line in the web graph output.

- [Adjusting Web Thresholds](#)
- [Creating a web summary](#)

Adjusting Web Thresholds

After you have created a web graph, you can adjust the thresholds controlling line styles using the toolbar slider to change the minimum visible line. You can also view additional threshold options by clicking the yellow double-arrow button on the toolbar to expand the web graph window. Then click the Controls tab to view additional options.

Figure 1. Expanded window featuring display and threshold options



Threshold values are. Shows the type of threshold selected during creation in the Web node dialog box.

Strong links are heavier. Selected by default, this is the standard way of viewing links between fields.

Weak links are heavier. Select to reverse the meaning of links displayed in bold lines. This option is frequently used for fraud detection or examination of outliers.

Web Display. Specify options for controlling the size of links in the output graph:

- **Size varies continuously.** Select to display a range of link sizes reflecting the variation in connection strengths based on actual data values.
- **Size shows strong/normal/weak.** Select to display three strengths of connections--strong, normal, and weak. The cutoff points for these categories can be specified above as well as in the final graph.

Strong links above. Specify a threshold for strong connections (heavy lines) and regular connections (normal lines). All connections above this value are considered strong. Use the slider to adjust the value or enter a number in the field.

Weak links below. Specify a number indicating the threshold for weak connections (dotted lines) and regular connections (normal lines). All connections below this value are considered weak. Use the slider to adjust the value or enter a number in the field.

After you have adjusted the thresholds for a web, you can replan, or redraw, the web display with the new threshold values through the web menu located on the web graph toolbar. Once you have found settings that reveal the most meaningful patterns, you can update the original settings in the Web node (also called the Parent Web node) by choosing Update Parent Node from the Web menu in the graph window.

Related information

- [Web Node](#)
- [Web Plot Tab](#)
- [Web Options Tab](#)
- [Web Appearance Tab](#)
- [Using a Web Graph](#)
- [Creating a web summary](#)

Creating a web summary

You can create a web summary document that lists strong, medium, and weak links by clicking the yellow double-arrow button on the toolbar to expand the web graph window. Then click the Summary tab to view tables for each type of link. Tables can be expanded and collapsed using the toggle buttons for each.

To print the summary, choose the following from the menu in the web graph window:

File>Print Summary

Related information

- [Web Node](#)
- [Web Plot Tab](#)
- [Web Options Tab](#)
- [Web Appearance Tab](#)
- [Using a Web Graph](#)
- [Adjusting Web Thresholds](#)

Evaluation node

The Evaluation node offers an easy way to evaluate and compare predictive models to choose the best model for your application. Evaluation charts show how models perform in predicting particular outcomes. They work by sorting records based on the predicted value and confidence of the prediction, splitting the records into groups of equal size (**quantiles**), and then plotting the value of the business criterion for each quantile, from highest to lowest. Multiple models are shown as separate lines in the plot.

Outcomes are handled by defining a specific value or range of values as a **hit**. Hits usually indicate success of some sort (such as a sale to a customer) or an event of interest (such as a specific medical diagnosis). You can define hit criteria on the Options tab of the dialog box, or you can use the default hit criteria as follows:

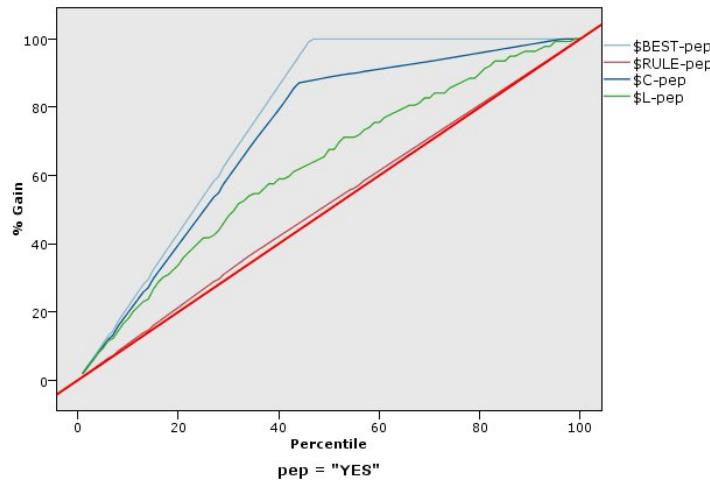
- Flag output fields are straightforward; hits correspond to *true* values.
- For Nominal output fields, the first value in the set defines a hit.
- For Continuous output fields, hits equal values greater than the midpoint of the field's range.

There are six types of evaluation charts, each of which emphasizes a different evaluation criterion.

Gains Charts

Gains are defined as the proportion of total hits that occurs in each quantile. Gains are computed as $(\text{number of hits in quantile} / \text{total number of hits}) \times 100\%$.

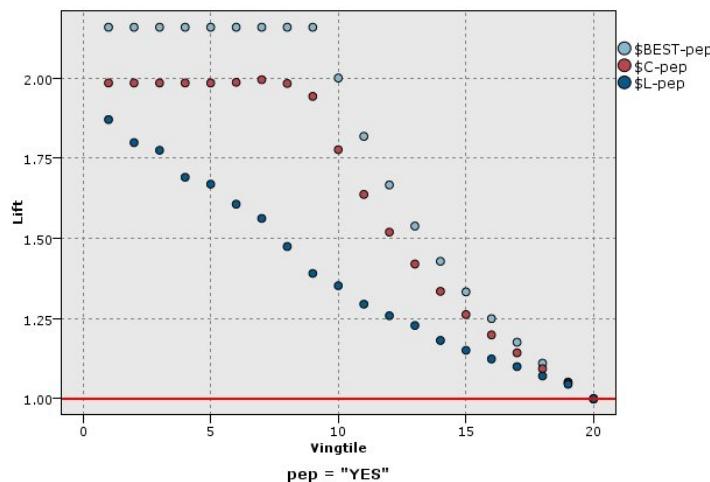
Figure 1. Gains chart (cumulative) with baseline, best line, and business rule displayed



Lift Charts

Lift compares the percentage of records in each quantile that are hits with the overall percentage of hits in the training data. It is computed as $(\text{hits in quantile} / \text{records in quantile}) / (\text{total hits} / \text{total records})$.

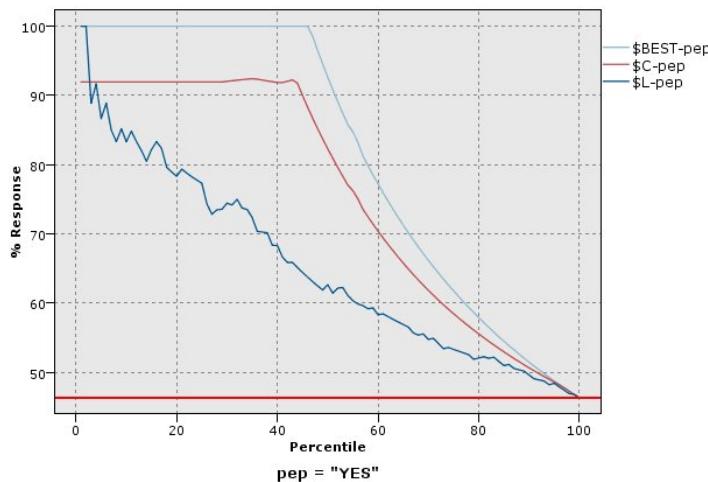
Figure 2. Lift chart (cumulative) using points and best line



Response Charts

Response is simply the percentage of records in the quantile that are hits. Response is computed as $(\text{hits in quantile} / \text{records in quantile}) \times 100\%$.

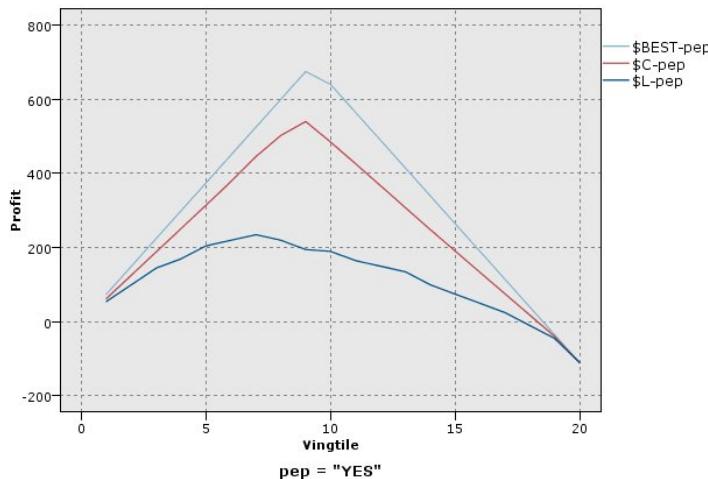
Figure 3. Response chart (cumulative) with best line



Profit Charts

Profit equals the revenue for each record minus the cost for the record. Profits for a quantile are simply the sum of profits for all records in the quantile. Revenues are assumed to apply only to hits, but costs apply to all records. Profits and costs can be fixed or can be defined by fields in the data. Profits are computed as (sum of revenue for records in quantile – sum of costs for records in quantile).

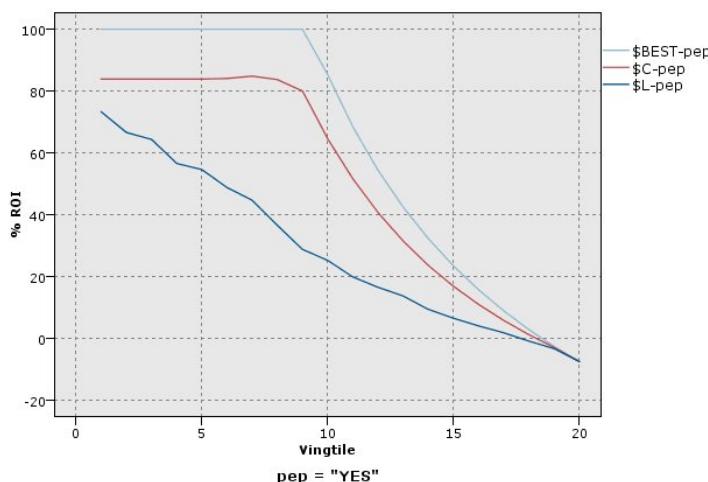
Figure 4. Profit chart (cumulative) with best line



ROI Charts

ROI (return on investment) is similar to profit in that it involves defining revenues and costs. ROI compares profits to costs for the quantile. ROI is computed as (profits for quantile / costs for quantile) × 100%.

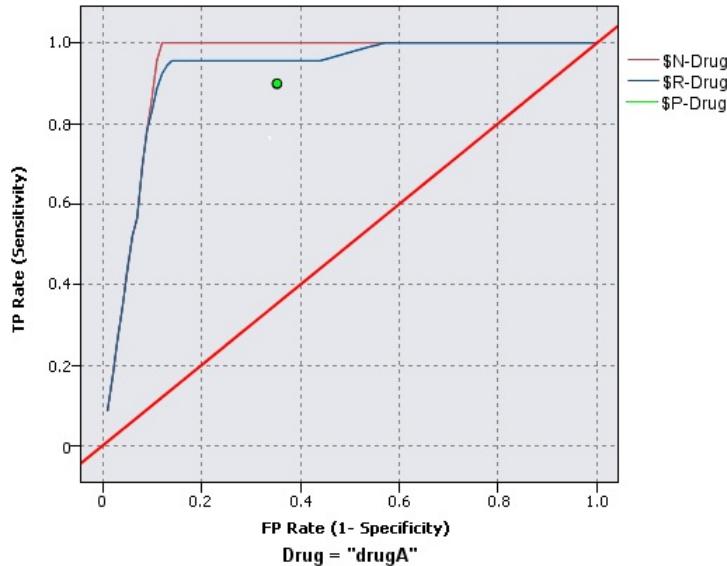
Figure 5. ROI chart (cumulative) with best line



ROC Charts

ROC (receiver operator characteristic) can only be used with binary classifiers. ROC can be used to visualize, organize and select classifiers based on their performance. A ROC chart plots the true positive rate (or sensitivity) against the false positive rate of the classifier. A ROC chart depicts the relative trade-offs between benefits (true positives) and costs (false positives). A true positive is an instance that is a hit and is classified as a hit. Therefore the true positive rate is calculated as the number of true positives / number of instances that are actually hits. A false positive is an instance that is a miss and is classified as a hit. Therefore the false positive rate is calculated as the number of false positives / number of instances that are actually misses.

Figure 6. ROC chart with best line



Evaluation charts can also be cumulative, so that each point equals the value for the corresponding quantile plus all higher quantiles. Cumulative charts usually convey the overall performance of models better, whereas noncumulative charts often excel at indicating particular problem areas for models.

Note: The Evaluation node doesn't support the use of commas in field names. If you have field names containing commas, you must either remove the commas or surround the field name in quotes.

- [Evaluation Plot Tab](#)
- [Evaluation Options tab](#)
- [Evaluation Appearance Tab](#)
- [Reading the Results of a Model Evaluation](#)
- [Using an Evaluation Chart](#)

Evaluation Plot Tab

Chart type. Select one of the following types: Gains, Response, Lift, Profit, ROI (return on investment), or ROC (receiver operator characteristic).

Cumulative plot. Select to create a cumulative chart. Values in cumulative charts are plotted for each quantile plus all higher quantiles. (Cumulative plot is not available for ROC charts.)

Include baseline. Select to include a baseline in the plot, indicating a perfectly random distribution of hits where confidence becomes irrelevant. (Include baseline is not available for Profit and ROI charts.)

Include best line. Select to include a best line in the plot, indicating perfect confidence (where hits = 100% of cases). (Include best line) is not available for ROC charts.)

Use profit criteria for all chart types. Select to use the profit criteria (cost, revenue, and weight) when calculating the evaluation measures, instead of the normal count of hits. For models with certain numeric targets, such as a model that predicts the revenue obtained from a customer in response to an offer, the value of the target field gives a better measure of the performance of the model than the count of hits. Selecting this option enables the Costs, Revenue, and Weight fields for Gains, Response, and Lift charts. To use the profit criteria for these three chart types, it is recommended that you set Revenue to be the target field, Cost to be 0.0 so that profit is equal to revenue, and that you specify a user defined hit condition of "true" so that all records are counted as hits. (Use profit criteria for all chart types is not available for ROC charts.)

Find predicted/predictor fields using. Select either Model output field metadata to search for the predicted fields in the graph using their metadata, or select Field name format to search for them by name.

Plot score fields. Select this check box to enable the score fields chooser. Then select one or more range, or continuous, score fields; that is, fields which are not strictly predictive models but which might be useful to rank records in terms of propensity to be a hit. The Evaluation node can compare any combination of one or more score fields with one or more predictive models. A typical example might be to compare several RFM fields with your best predictive model.

Target. Select the target field using the field chooser. Choose any instantiated flag or nominal field with two or more values.

Note: This target field is only applicable to score fields (predictive models define their own targets), and is ignored if a custom hit criterion is set on the Options tab.

Split by partition. If a partition field is used to split records into training, test, and validation samples, select this option to display a separate evaluation chart for each partition. See the topic [Partition Node](#) for more information.

Note: When splitting by partition, records with null values in the partition field are excluded from the evaluation. This will never be an issue if a Partition node is used, since Partition nodes do not generate null values.

Plot. Select the size of quantiles to plot in the chart from the drop-down list. Options include Quartiles, Quintiles, Deciles, Vingtiles, Percentiles, and 1000-tiles. (Plot is not available for ROC charts.)

Style. Select Line or Point.

For all chart types except ROC charts, additional controls enable you to specify costs, revenue, and weights.

- Costs. Specify the cost associated with each record. You can select Fixed or Variable costs. For fixed costs, specify the cost value. For variable costs, click the Field Chooser button to select a field as the cost field. (Costs is not available for ROC charts.)
- Revenue. Specify the revenue associated with each record that represents a hit. You can select Fixed or Variable costs. For fixed revenue, specify the revenue value. For variable revenue, click the Field Chooser button to select a field as the revenue field. (Revenue is not available for ROC charts.)
- Weight. If the records in your data represent more than one unit, you can use frequency weights to adjust the results. Specify the weight associated with each record, using Fixed or Variable weights. For fixed weights, specify the weight value (the number of units per record). For variable weights, click the Field Chooser button to select a field as the weight field. (Weight is not available for ROC charts.)

Related information

- [Evaluation node](#)
 - [Evaluation Options tab](#)
 - [Evaluation Appearance Tab](#)
 - [Reading the Results of a Model Evaluation](#)
 - [Using an Evaluation Chart](#)
-

Evaluation Options tab

The Options tab for evaluation charts provides flexibility in defining hits, scoring criteria, and business rules displayed in the chart. You can also set options for exporting the results of the model evaluation.

User defined hit. Select to specify a custom condition used to indicate a hit. This option is useful for defining the outcome of interest rather than deducing it from the type of target field and the order of values.

- Condition. When User defined hit is selected above, you must specify a CLEM expression for a hit condition. For example, `@TARGET = "YES"` is a valid condition indicating that a value of Yes for the target field will be counted as a hit in the evaluation. The specified condition will be used for all target fields. To create a condition, type in the field or use the Expression Builder to generate a condition expression. If the data are instantiated, you can insert values directly from the Expression Builder.

User defined score. Select to specify a condition used for scoring cases before assigning them to quantiles. The default score is calculated from the predicted value and the confidence. Use the Expression field to create a custom scoring expression.

- Expression. Specify a CLEM expression used for scoring. For example, if a numeric output in the range 0–1 is ordered so that lower values are better than higher, you might define a hit as `@TARGET < 0.5` and the associated score as `1 - @PREDICTED`. The score expression must result in a numeric value. To create a condition, type in the field or use the Expression Builder to generate a condition expression.

Include business rule. Select to specify a rule condition reflecting criteria of interest. For example, you may want to display a rule for all cases where `mortgage = "Y" and income >= 33000`. Business rules are drawn on the chart and labeled in the key as *Rule*. (Include business rule is not supported for ROC charts.)

- Condition. Specify a CLEM expression used to define a business rule in the output chart. Simply type in the field or use the Expression Builder to generate a condition expression. If the data are instantiated, you can insert values directly from the Expression Builder.

Export results to file. Select to export the results of the model evaluation to a delimited text file. You can read this file to perform specialized analyses on the calculated values. Set the following options for export:

- **Filename.** Enter the filename for the output file. Use the ellipsis button (...) to browse to the desired folder.
- **Delimiter.** Enter a character, such as a comma or space, to use as the field delimiter.

Include field names. Select this option to include field names as the first line of the output file.

New line after each record. Select this option to begin each record on a new line.

Evaluation Appearance Tab

You can specify appearance options before graph creation.

Title. Enter the text to use for the graph's title.

Subtitle. Enter the text to use for the graph's subtitle.

Text. Either accept the automatically generated text label, or select Custom to specify a label.

X label. Either accept the automatically generated x-axis (horizontal) label or select Custom to specify a label.

Y label. Either accept the automatically generated y-axis (vertical) label or select Custom to specify a label.

Display gridline. Selected by default, this option displays a gridline behind the plot or graph that enables you to more easily determine region and band cutoff points. Gridlines are always displayed in white unless the graph background is white; in this case, they are displayed in gray.

Related information

- [Evaluation node](#)
- [Evaluation Plot Tab](#)
- [Evaluation Options tab](#)
- [Reading the Results of a Model Evaluation](#)
- [Using an Evaluation Chart](#)

Reading the Results of a Model Evaluation

The interpretation of an evaluation chart depends to a certain extent on the type of chart, but there are some characteristics common to all evaluation charts. For cumulative charts, higher lines indicate better models, especially on the left side of the chart. In many cases, when comparing multiple models the lines will cross, so that one model will be higher in one part of the chart and another will be higher in a different part of the chart. In this case, you need to consider what portion of the sample you want (which defines a point on the x axis) when deciding which model to choose.

Most of the noncumulative charts will be very similar. For good models, noncumulative charts should be high toward the left side of the chart and low toward the right side of the chart. (If a noncumulative chart shows a sawtooth pattern, you can smooth it out by reducing the number of quantiles to plot and re-executing the graph.) Dips on the left side of the chart or spikes on the right side can indicate areas where the model is predicting poorly. A flat line across the whole graph indicates a model that essentially provides no information.

Gains charts. Cumulative gains charts always start at 0% and end at 100% as you go from left to right. For a good model, the gains chart will rise steeply toward 100% and then level off. A model that provides no information will follow the diagonal from lower left to upper right (shown in the chart if Include baseline is selected).

Lift charts. Cumulative lift charts tend to start above 1.0 and gradually descend until they reach 1.0 as you go from left to right. The right edge of the chart represents the entire dataset, so the ratio of hits in cumulative quantiles to hits in data is 1.0. For a good model, lift should start well above 1.0 on the left, remain on a high plateau as you move to the right, and then trail off sharply toward 1.0 on the right side of the chart. For a model that provides no information, the line will hover around 1.0 for the entire graph. (If Include baseline is selected, a horizontal line at 1.0 is shown in the chart for reference.)

Response charts. Cumulative response charts tend to be very similar to lift charts except for the scaling. Response charts usually start near 100% and gradually descend until they reach the overall response rate (total hits / total records) on the right edge of the chart. For a good model, the line will start near or at 100% on the left, remain on a high plateau as you move to the right, and then trail off sharply toward the overall response rate on the right side of the chart. For a model that provides no information, the line will hover around the overall response rate for the entire graph. (If Include baseline is selected, a horizontal line at the overall response rate is shown in the chart for reference.)

Profit charts. Cumulative profit charts show the sum of profits as you increase the size of the selected sample, moving from left to right. Profit charts usually start near 0, increase steadily as you move to the right until they reach a peak or plateau in the middle, and then decrease toward

the right edge of the chart. For a good model, profits will show a well-defined peak somewhere in the middle of the chart. For a model that provides no information, the line will be relatively straight and may be increasing, decreasing, or level depending on the cost/revenue structure that applies.

ROI charts. Cumulative ROI (return on investment) charts tend to be similar to response charts and lift charts except for the scaling. ROI charts usually start above 0% and gradually descend until they reach the overall ROI for the entire dataset (which can be negative). For a good model, the line should start well above 0%, remain on a high plateau as you move to the right, and then trail off rather sharply toward the overall ROI on the right side of the chart. For a model that provides no information, the line should hover around the overall ROI value.

ROC charts. ROC curves generally have the shape of a cumulative gains chart. The curve starts at the (0,0) coordinate and ends at the (1,1) coordinate as you go from left to right. A chart that rises steeply toward the (0,1) coordinate then levels off indicates a good classifier. A model that classifies instances at random as hits or misses will follow the diagonal from lower left to upper right (shown in the chart if Include baseline is selected). If no confidence field is provided for a model, the model is plotted as a single point. The classifier with the optimum threshold of classification is located closest to the (0,1) coordinate, or upper left corner, of the chart. This location represents a high number of instances that are correctly classified as hits, and a low number of instances that are incorrectly classified as hits. Points above the diagonal line represent good classification results. Points below the diagonal line represent poor classification results that are worse than if the instances were classified at random.

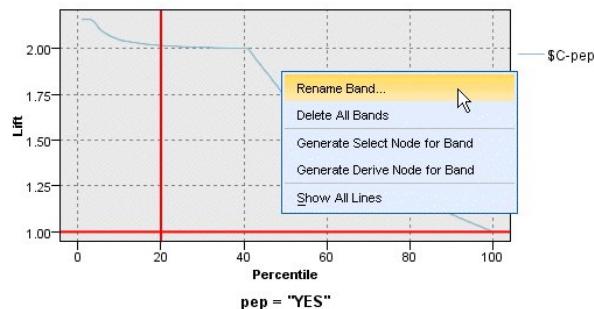
Related information

- [Evaluation node](#)
- [Evaluation Plot Tab](#)
- [Evaluation Options tab](#)
- [Evaluation Appearance Tab](#)
- [Using an Evaluation Chart](#)

Using an Evaluation Chart

Using the mouse to explore an evaluation chart is similar to using a histogram or collection graph. The x axis represents model scores across the specified quantiles, such as vingtiles or deciles.

Figure 1. Working with an evaluation chart



You can partition the x axis into bands just as you would for a histogram by using the splitter icon to display options for automatically splitting the axis into equal bands. See the topic [Exploring Graphs](#) for more information. You can manually edit the boundaries of bands by selecting Graph Bands from the Edit menu.

Once you have created an evaluation chart, defined bands, and examined the results, you can use options on the Generate menu and the context menu to automatically create nodes based upon selections in the graph. See the topic [Generating Nodes from Graphs](#) for more information.

When generating nodes from an evaluation chart, you will be prompted to select a single model from all available models in the chart.

Select a model and click OK to generate the new node onto the stream canvas.

Related information

- [Evaluation node](#)
- [Evaluation Plot Tab](#)
- [Evaluation Options tab](#)
- [Evaluation Appearance Tab](#)
- [Reading the Results of a Model Evaluation](#)
- [Editing Visualizations](#)

Map Visualization Node

The Map Visualization node can accept multiple input connections and display geospatial data on a map as a series of layers. Each layer is a single geospatial field; for example, the base layer might be a map of a country, then above that you might have one layer for roads, one layer for rivers, and one layer for towns.

Although most geospatial datasets ordinarily contain a single geospatial field, when there are multiple geospatial fields in a single input you can choose which fields to display. Two fields from the same input connection cannot be displayed at the same time; however, you can copy and paste the incoming connection and display a different field from each.

- [Map Visualization Plot Tab](#)
- [Map Visualization Appearance Tab](#)

Related information

- [Map Visualization Plot Tab](#)
- [Map Visualization Appearance Tab](#)
- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Plot Node](#)
- [Multiplot Node](#)
- [Time Plot Node](#)
- [Distribution Node](#)
- [Histogram Node](#)
- [Collection Node](#)
- [Web Node](#)
- [Evaluation node](#)
- [Working with Graph Output](#)
- [Exploring Graphs](#)
- [Editing Visualizations](#)
- [Adding Titles and Footnotes](#)
- [Using Graph Stylesheets](#)
- [Printing, saving, copying, and exporting graphs](#)

Map Visualization Plot Tab

Layers

This table displays information about the inputs to the map node. The order of the layers dictates the order in which the layers are displayed in both the map preview and the visual output when the node is executed. The top row in the table is the 'top' layer and the bottom row is the 'base' layer; in other words, each layer is shown on the map in front of the layer directly below it in the table.

Note: Where a layer in the table contains a three-dimensional geospatial field only the x and y axes are plotted. The z axis is ignored.

Name

Names are automatically created for each layer and are made up using the following format: `tag[source_node:connected_node]`. By default, the tag is shown as a number, with 1 representing the first input that is connected, 2 for the second input, and so on. If required, press the Edit Layer button to change the tag in the Change Map Layer Options dialog box. For example, you might change the tag to be "roads" or "cities" to reflect the data input.

Type

Shows the measurement type icon of the geospatial field that is selected as the layer. If the input data contains multiple fields with a geospatial measurement type the default selection uses the following sort order:

1. Point
2. Linestring
3. Polygon
4. MultiPoint
5. MultiLinestring
6. MultiPolygon

Note: If there are two fields with the same measurement type, the first field (alphabetically by name) is selected by default.

Symbol

Note: This column is only completed for Point and MultiPoint fields.

Shows the symbol that is used for Point or MultiPoint fields. If required, press the Edit Layer button to change the symbol in the Change Map Layer Options dialog box.

Color

Shows the color that is selected to represent layer on the map. If required, press the Edit Layer button to change the color in the Change Map Layer Options dialog box. The color is applied to different items depending on the measurement type.

- For Points or MultiPoints, the color is applied to the symbol for the layer.
- For Linestrings and Polygons, the color is applied to the entire shape. Polygons always have a black outline; the color that is shown in the column is the color that is used to fill the shape.

Preview

This pane shows a preview of the current selection of inputs in the Layers table. The preview takes into account the order of layers, symbol, color, and any other display settings that are associated with the layers and, when possible, updates the display whenever the settings change. If you change details elsewhere in your stream, such as the geospatial fields to use as layers, or if you amend details such as associated aggregation functions, you might need to click the Refresh Data button to update the preview.

Use the Preview to set your display settings before you run your stream. To protect against time delays that could be caused by using a large dataset, the preview samples each layer and creates a display from the first 100 records.

- [Change map layers](#)

Related information

- [Map Visualization Node](#)
 - [Map Visualization Appearance Tab](#)
 - [Change map layers](#)
-

Change map layers

You can use the Change Map Layer Options dialog box to amend various details of any layer that is shown on the Plot tab of the Map Visualization node.

Input details

Tag

By default, the tag is a number; you can replace this number with a more meaningful tag to help identify the layer on the map. For example, the tag might be the name of the data input, such as "Cities".

Layer field

If you have more than one geospatial field in your input data, use this option to select the field that you want to display as a layer on the map.

By default, the layers you can choose from are in the following sort order.

- Point
- LineString
- Polygon
- MultiPoint
- MultiLineString
- MultiPolygon

Display settings

Use hex binning

Note: This option affects only point and multipoint fields.

Hex (hexagonal) binning combines proximate points (based on their x and y coordinates) into a single point to display on the map. The single point is shown as a hexagon, but is effectively rendered as a polygon.

Because the hexagon is rendered as a polygon, any point fields with hex binning turned on are treated as polygons. This means that if you choose to Order by type on the map node dialog box, any point layers that have hex binning applied are rendered above polygon layers but below linestring and point layers.

If you use hex binning for a multipoint field, the field is first converted into a point field by binning the multipoint values to calculate the central point. Central points are used to calculate the hex bins.

Aggregation

Note: This column is only available when you select the Use hex binning check box and also select an Overlay.

If you select an Overlay field for a points layer that uses hex binning, all of the values in that field must be aggregated for all points within the hexagon. Specify an aggregation function for any overlay fields you want to apply to the map. The available aggregation functions depend on the measurement type.

- Aggregation functions for a Continuous measurement type, with Real or Integer storage:
 - Sum
 - Mean
 - Min
 - Max
 - Median
 - First Quartile
 - Third Quartile
- Aggregation functions for a Continuous measurement type, with Time, Date, or Timestamp storage:
 - Mean
 - Min
 - Max
- Aggregation functions for Nominal or Categorical measurement types:
 - Mode
 - Min
 - Max
- Aggregation functions for a Flag measurement type:
 - True, if any are true
 - False, if any are false

Color

Use this option to choose either a standard color to apply to all features of the geospatial field, or an overlay field, which colors the features based on the values from another field in the data.

If you select Standard, you can choose a color from the palette of colors that are shown in the Chart Category Color Order pane in the Display tab of the User Options dialog box.

For more information, see [Setting display options](#).

If you select Overlay, you can choose any field from the data source that contains the geospatial field that was selected as the Layer field.

- For nominal or categorical overlay fields, the color palette that you can choose from is the same as that shown for the Standard color options.
- For continuous and ordinal overlay fields a second drop-down list is displayed, from which you select a color. When you select a color, the overlay is applied by varying the saturation of that color according to the values in the continuous or ordinal field. The highest value uses the color chosen from the drop-down list and lower values are shown by correspondingly lower saturations.

Symbol

Note: Only enabled for point and multipoint measurement types.

Use this option to choose whether to have a Standard symbol, which is applied to all records of your geospatial field, or an Overlay symbol, which changes the symbol icon for the points based on the values of another field in your data.

If you select Standard, you can choose one of the default symbols from a drop-down list to represent the point data on the map.

If you select Overlay, you can choose any nominal, ordinal, or categorical field from the data source that contains the geospatial field that was selected as the Layer field. For each value in the overlay field, a different symbol is displayed on the map.

For example, your data might contain a point field that represents the locations of shops and the overlay might be a shop type field. In this example, all food shops might be identified on the map by a cross symbol and all electronics shops by a square symbol.

Size

Note: Only enabled for point, multipoint, linestring and multilinestring measurement types.

Use this option to choose whether to have a Standard size, which is applied to all records of your geospatial field, or an Overlay size, which changes the size of the symbol icon or line thickness based on the values of another field in your data.

If you select Standard, you can choose a pixel width value. Options available are 1, 2, 3, 4, 5, 10, 20, or 30.

If you select Overlay, you can choose any field from the data source that contains the geospatial field that was selected as the Layer field. The thickness of the line or point varies depending on the value of the chosen field.

Transparency

Use this option to choose whether to have a Standard transparency, which is applied to all records of your geospatial field, or an Overlay transparency, which changes the transparency of the symbol, line, or polygon based on the values of another field in your data.

If you select Standard, you can choose from a selection of transparency levels that start from 0% (opaque) and increase in 10% increments to 100% (transparent).

If you select Overlay, you can choose any field from the data source that contains the geospatial field that was selected as the Layer field. A different level of transparency is displayed on the map for each value in the overlay field. The transparency is applied to the color chosen from the color drop-down list for the point, line, or polygon.

Data Label

Note: This option is not available if you select the Use hex binning check box.

Use this option to select a field to use as data labels on the map. For example, if applied to a polygon layer, the data label might be the name field, containing the name of each polygon. If you select the name field, those names are displayed on the map.

Related information

- [Map Visualization Node](#)
 - [Map Visualization Appearance Tab](#)
 - [Map Visualization Plot Tab](#)
-

Map Visualization Appearance Tab

You can specify appearance options before graph creation.

Title. Enter the text to use for the graph's title.

Subtitle. Enter the text to use for the graph's subtitle.

Caption. Enter the text to use for the graph's caption.

Related information

- [Map Visualization Node](#)
 - [Map Visualization Plot Tab](#)
-

t-SNE node

t-Distributed Stochastic Neighbor Embedding (t-SNE)© is a tool for visualizing high-dimensional data. It converts affinities of data points to probabilities. The affinities in the original space are represented by Gaussian joint probabilities and the affinities in the embedded space are represented by Student's t-distributions. This allows t-SNE to be particularly sensitive to local structure and has a few other advantages over existing techniques:¹

- Revealing the structure at many scales on a single map
- Revealing data that lie in multiple, different, manifolds, or clusters
- Reducing the tendency to crowd points together at the center

The t-SNE node in SPSS® Modeler is implemented in Python and requires the scikit-learn© Python library. For details about t-SNE and the scikit-learn library, see:

- <https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html#sklearn.manifold.TSNE>
- <https://scikit-learn.org/stable/modules/manifold.html#t-sne>

The Python tab on the Nodes Palette contains this node and other Python nodes. The t-SNE node is also available on the Graphs tab.

¹ References:

van der Maaten, L.J.P.; Hinton, G. ["Visualizing High-Dimensional Data using t-SNE."](#) Journal of Machine Learning Research. 9:2579-2605, 2008.

van der Maaten, L.J.P. ["t-Distributed Stochastic Neighbor Embedding."](#)

van der Maaten, L.J.P. ["Accelerating t-SNE using Tree-Based Algorithms."](#) Journal of Machine Learning Research. 15(Oct):3221-3245, 2014.

- [t-SNE node Expert options](#)
- [t-SNE node Output options](#)
- [Accessing and plotting t-SNE data](#)
- [t-SNE model nuggets](#)
- [t-SNE node Expert options](#)
- [t-SNE node Output options](#)
- [t-SNE model nuggets](#)

t-SNE node Expert options

Choose Simple mode or Expert mode depending on which options you want to set for the t-SNE node.

Visualization type. Select 2D or 3D to specify whether to draw the graph as two-dimensional or three-dimensional.

Method. Select Barnes Hut or Exact. By default, the gradient calculation algorithm uses Barnes-Hut approximation which runs much faster than the Exact method. Barnes-Hut approximation allows the t-SNE technique to be applied to large, real-world datasets. The Exact algorithm will do a better job of avoiding nearest-neighbor errors.

Init. Select Random or PCA for the initialization of embedding.

Target Field. Select the target field to show as a colormap on the output graph. The graph will use one color if no target field is specified here.

Optimization

Perplexity. The perplexity is related to the number of nearest neighbors that are used in other manifold learning algorithms. Larger datasets usually require a larger perplexity. Consider selecting a value between 5 and 50. Default is 30, and the range is 2 - 9999999.

Early exaggeration. This setting controls how tight natural clusters in the original space will be in the embedded space, and how much space will be between them. Default is 12, and the range is 2 - 9999999.

Learning rate. If the learning rate is too high, the data might look like a "ball" with any point approximately equidistant from its nearest neighbors. If the learning rate is too low, most points may look compressed in a dense cloud with few outliers. If the cost function gets stuck in a bad local minimum, increasing the learning rate may help. Default is 200, and the range is 0 - 9999999.

Max iterations. The maximum number of iterations for the optimization. Default is 1000, and the range is 250 - 9999999.

Angular size. The angular size of a distant node as measured from a point. Enter a value between 0 and 1. Default is 0.5.

Random seed

Set random seed. Select this option and click Generate to generate the seed used by the random number generator.

Optimization stop condition

Max iterations without progress. The maximum number of iterations without progress to perform before stopping the optimization, used after 250 initial iterations with early exaggeration. Note that progress is only checked every 50 iterations, so this value is rounded to the next multiple of 50. Default is 300, and the range is 0 - 9999999.

Min gradient norm. If the gradient norm is below this minimum threshold, the optimization will stop. Default is 1.0E-7.

Metric. The metric to use when calculating distance between instances in a feature array. If the metric is a string, it must be one of the options allowed by `scipy.spatial.distance.pdist` for its metric parameter, or a metric listed in `pairwise.PAIRWISE_DISTANCE_FUNCTIONS`. Select one of the available metric types. Default is euclidean.

When number of records greater than. Specify a method for plotting large datasets. You can specify a maximum dataset size or use the default 2,000 points. Performance is enhanced for large datasets when you select the Bin or Sample options. Alternatively, you can choose to plot all data points by selecting Use all data, but you should note that this may dramatically decrease the performance of the software.

- Bin. Select to enable binning when the dataset contains more than the specified number of records. Binning divides the graph into fine grids before actually plotting and counts the number of connections that would appear in each of the grid cells. In the final graph, one connection is used per cell at the bin centroid (average of all connection points in the bin).
- Sample. Select to randomly sample the data to the specified number of records.

The following table shows the relationship between the settings on the Expert tab of the SPSS® Modeler t-SNE node dialog and the Python t-SNE library parameters.

Table 1. Node properties mapped to Python library parameters

SPSS Modeler setting	Script name (property name)	Python t-SNE parameter
Mode	<code>mode_type</code>	
Visualization type	<code>n_components</code>	<code>n_components</code>
Method	<code>method</code>	<code>method</code>
Initialization of embedding	<code>init</code>	<code>init</code>
Target	<code>target_field</code>	<code>target_field</code>
Perplexity	<code>perplexity</code>	<code>perplexity</code>
Early exaggeration	<code>early_exaggeration</code>	<code>early_exaggeration</code>

SPSS Modeler setting	Script name (property name)	Python t-SNE parameter
Learning rate	learning_rate	learning_rate
Max iterations	n_iter	n_iter
Angular size	angle	angle
Set random seed	enable_random_seed	
Random seed	random_seed	random_state
Max iterations without progress	n_iter_without_progress	n_iter_without_progress
Min gradient norm	min_grad_norm	min_grad_norm
Perform t-SNE with multiple perplexities	isGridSearch	

t-SNE node Output options

Specify options for the t-SNE node output on the Output tab.

Output name. Specify the name of the output that is produced when the node runs. If you select Auto, the name of the output is automatically set.

Output to screen. Select this option to generate and display the output in a new window. The output is also added to the Output manager.

Output to file. Select this option to save the output to a file. Doing so enables the File name and File type fields. The t-SNE node requires access to this output file if you want to create plots using other fields for comparison purposes – or to use its output as predictors in classification or regression models. The t-SNE model creates a result file of x, y (and z) coordinate fields that is most easily accessed using a Fixed File source node.

Accessing and plotting t-SNE data

If you use the Output to file option to save t-SNE output to files, you can then create plots using other fields for comparison purposes – or use the output as predictors in classification or regression models. The t-SNE model creates a result file of x, y (and z) coordinate fields that is most easily accessed using a Fixed File source node. This section provides example information.

1. In the t-SNE node dialog, open the Output tab.
 2. Select Output to file and type a file name. Use the default HTML file type. When you run the model, this will generate three output files in your output location:
 - A text file (result_xxxxxx.txt)
 - An HTML file (the file name you specified)
 - A PNG file (tsne_chart_yyyyyy.png)
- The text file will contain the data you need, but for technical reasons it may be in standard or scientific format. If it's in scientific format (1.1111111e+01), then you need to create a new stream that recognizes the format:

Accessing t-SNE plot data when the text file is in scientific numeric format

1. Create a new stream (File > New Stream).
2. Go to Tools > Stream Properties > Options, select Number formats, and select Scientific (#.###E+##) for the number display format.
3. Add a Fixed File source node to your canvas, and use the following settings on the File tab:
 - Skip header lines: 1
 - Record length: 54
 - tSNE_x Start: 3, Length: 16
 - tSNE_y Start: 20, Length: 16
 - tSNE_z Start: 36, Length: 16
4. On the Type tab, the numbers should be recognized as Real. Click Read Values and you should see field values similar to:

Table 1. Example field values

Field	Measurement	Values
tSNE_x	Continuous	[-7.07176703, 7.14338837]
tSNE_y	Continuous	[-9.2188112, 8.89647667]
tSNE_x	Continuous	[-9.95892882, 9.95742482]

5. Add a Select node to the stream so you can delete the following bottom two rows of text in the file which are read as nulls:

```
*****
Perform t-SNE (total time 9.5s)
```

On the Settings tab of the Select node, select Discard for the mode and use the condition @NULL(tSNE_x) to delete the rows.

6. Add a Type node and a Flat File export node to the stream to create a Var. File source node that will copy and paste back into your original stream.

Accessing t-SNE plot data when the text file is in standard numeric format

1. Create a new stream (File > New Stream).
2. Add a Fixed File source node to your canvas. The following three nodes are all that's required to access the t-SNE data.

Figure 1. Stream for accessing t-SNE plot data in standard numeric format



3. Use the following settings on the File tab of the Fixed File source node:
 - Skip header lines: 1
 - Record length: 29
 - tSNE_x Start: 3, Length: 12
 - tSNE_y Start: 16, Length: 12
4. On the Filter tab, you can rename **field1** and **field2** to **tsneX** and **tsneY**.
5. Add a Merge node to connect it to your stream by using the Order merge method.
6. You can now use a Plot node to plot **tsneX** versus **tsneY** and color it with your field under investigation.

t-SNE model nuggets

t-SNE model nuggets contain all of the information captured by the t-SNE model. The following tabs are available.

Graph

The Graph tab displays chart output for the t-SNE node. A pyplot scatter chart shows the low dimensions result. If you didn't select the Perform t-SNE with multiple perplexities option on the [Expert](#) tab of the t-SNE node, only one graph is included rather than six graphs with different perplexities.

Text output

The Text output tab displays the results of the t-SNE algorithm. If you chose the 2D visualization type on the [Expert](#) tab of the t-SNE node, the result here is the point value in two dimensions. If you chose 3D, the result is the point value in three dimensions.

E-Plot (Beta) node

E-Plot (Beta) nodes show the relationship between numeric fields. The E-Plot (Beta) node is similar to the [Plot](#) node, but its options differ and it uses new graphing capabilities. Use the node to play around with new graphing features in SPSS® Modeler.

The E-Plot (Beta) node provides scatterplots, line charts, and bar charts to illustrate the relationship between numeric fields. The new graphing interface in this node is intuitive and modern, very customizable, and the data charts are interactive. For more information, see [Using an e-plot graph](#).

- [E-Plot \(Beta\) node Plot tab](#)
- [E-Plot \(Beta\) node Options tab](#)
- [E-Plot \(Beta\) Appearance tab](#)
- [Using an e-plot graph](#)

E-Plot (Beta) node Plot tab

Plots show values of a Y field against values of an X field. Often, these fields correspond to a dependent variable and an independent variable, respectively.

X field. From the list, select the field to display on the horizontal x axis.

Y field. From the list, select the field to display on the vertical y axis.

Overlay. There are different ways to illustrate categories for data values. For example, you might use a *maincrop* field as a color overlay to indicate the *estincome* and *claimvalue* values for the main crop grown by claim applicants. Select fields for color mapping, size mapping, and shape mapping in the output. Also select any other interested fields to include in the interactive output. See the topic [Aesthetics, Overlays, Panels, and Animation](#) for more information.

Once you have set options for an e-plot, you can run the plot directly from the dialog box by clicking Run. You may, however, want to use the Options tab for additional specifications.

E-Plot (Beta) node Options tab

Maximum number of records to plot. Specify a method for plotting large datasets. You can specify a maximum dataset size or use the default 2,000 records. Performance is enhanced for large datasets when you select the Sample option. The Sample option randomly samples the data to the number of records entered in the text field. Alternatively, you can choose to plot all data points by selecting Use all data, but you should note that this may dramatically decrease the performance of the software.

E-Plot (Beta) Appearance tab

You can specify a title and subtitle before graph creation, if desired. These options can also be specified or changed after graph creation.

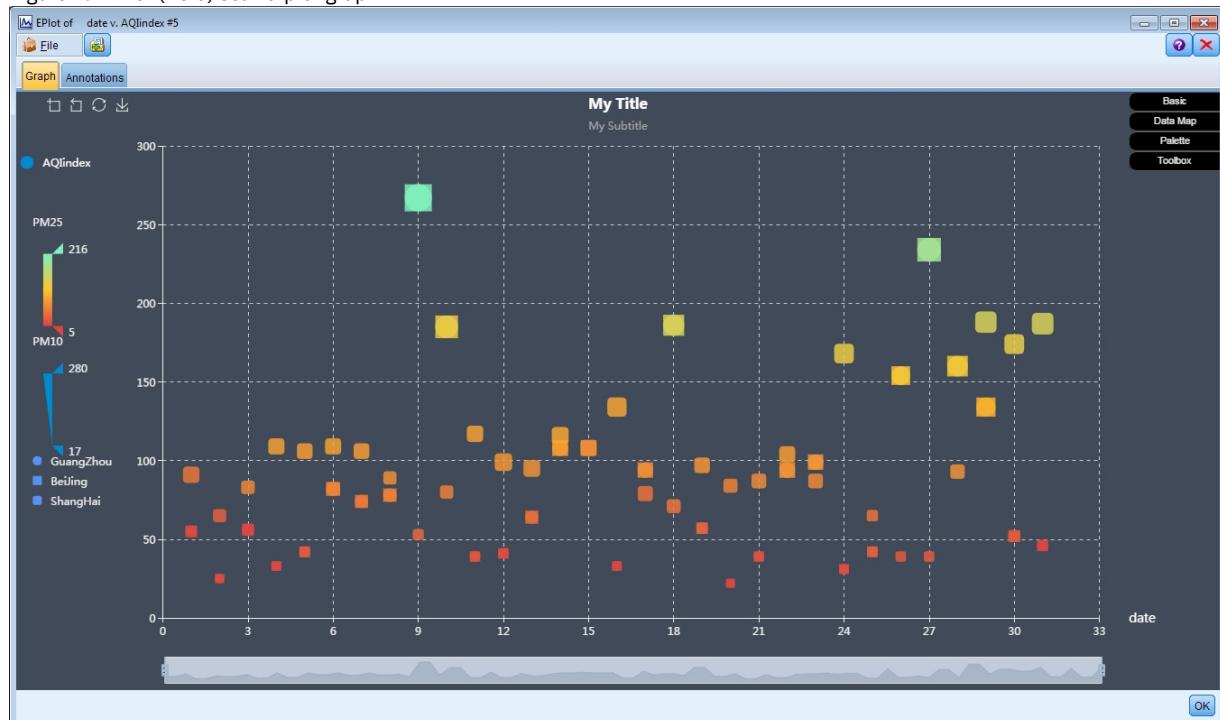
Title. Enter the text to use for the graph's title.

Subtitle. Enter the text to use for the graph's subtitle.

Using an e-plot graph

The E-Plot (Beta) node provides scatterplots, line charts, and bar charts to illustrate the relationship between numeric fields. The new graphing interface introduced in this beta node includes many new and improved capabilities.

Figure 1. E-Plot (Beta) scatterplot graph



On the Graph tab, the upper left corner provides a toolbar for zooming in on a specific section of the chart, zooming back out, returning to the initial full view, or saving the chart for external use:

Figure 2. Toolbar



At the bottom of the window, you can use the slider to zoom in on a specific section of the chart. Move the small rectangular controls on the right and the left to zoom. To use this slider, you must first turn it on in the Toolbox options area.

Figure 3. Zoom slider



The left side of the window provides controls for changing the range of values shown. To use these controls, you must first specify options in the Data Map options area. In the example below, a field named **PM25** is selected for the color map, a field named **PM10** is selected for the size map, and a field called **City** is selected for the shape map. You can hover over the vertical color bars to highlight the corresponding areas of the graph, or slide the upper and lower triangles.

Figure 4. Range controls



On the right side of the window, a set of expandable options is available that you can use to interact with the data and change the chart's appearance in real time:

Figure 5. Expandable options



Basic options

Figure 6. Basic options	Select the dark or light theme, specify a title and a subtitle, select a chart type (scatter, line, or bar), and select the series shown on the Y axis. If you select Line chart, only the fields on the Y axis will be displayed and only the fields on the Y axis will be available in the Data Map options for color map and size map. If you select Bar chart, only color map options will be available in the Data Map options. For the series, all Interested fields you selected on the Plot tab of the E-Plot node will be available here.
-------------------------	--

Data Map options

Figure 7. Data map options 	<p>Select a continuous field or a categorical field for the Color Map. If a continuous field is selected, colors from green to red will be shown. The lower the value is, the closer to red its color will be – and the higher the value is, the closer to green its color will be. If a categorical field is selected, the field color will be displayed according to the defined color palette.</p> <p>Size Map only supports continuous fields. The lower the value is on the chart, the smaller its plot size will be.</p> <p>Shape Map only supports categorical fields. The shape displayed on the map is defined by a categorical field that splits the visualization into elements of different shapes – one for each category.</p>
--	---

Palette options

Figure 8. Palette options 	<p>Use the Palette if you want to customize the colors used for the title and the series. Select the title or the series from the drop-down, click Edit Predefined Colors, and then click More to select a color. Or you can use the RGB or Hex fields to specify an exact color.</p>
---	---

Toolbox options

Figure 9. Toolbox options 	<p>Use the Toolbox options to turn on or off the zoom slider, set gridline properties, and turn on or off mouse tracking. Mouse tracking shows your exact coordinate position when hovering over the chart.</p>
---	---

Working with Graph Output

When you run a stream and obtain a graph, you can interact and edit this graph directly in the output window. For example, you can:

- [Explore the graph](#) by analyzing the data and identifying values. You can draw bands and regions or mark elements to generate Select, Derive, or Balance nodes
- [Edit the graph](#) to change the graph's layout and look.
- [Apply title, footnote, or axis labels](#).
- [Update or change the stylesheet](#).
- [Print, save, copy, or export the graph](#).

Related information

- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Plot Node](#)
- [Distribution Node](#)
- [Histogram Node](#)
- [Collection Node](#)
- [Multiplot Node](#)
- [Web Node](#)
- [Time Plot Node](#)
- [Evaluation node](#)
- [Exploring Graphs](#)
- [Editing Visualizations](#)
- [Adding Titles and Footnotes](#)
- [Using Graph Stylesheets](#)
- [Printing, saving, copying, and exporting graphs](#)

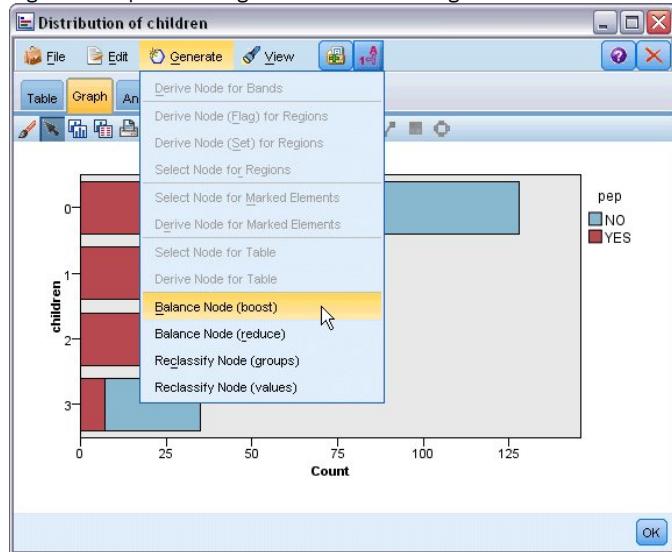
Exploring Graphs

While Edit mode allows you to edit the graph's layout and look, Explore mode allows you to analytically explore the data and values represented by the graph. The main goal of exploration is to analyze the data and then identify values using bands, regions, and marking to generate Select, Derive, or Balance nodes. To select this mode, choose View > Explore Mode from the menus (or click the toolbar icon).

While some graphs can use all of the exploration tools, others accept only one. Explore mode includes:

- Defining and editing bands, which are used to split the values along a scale x axis. See the topic [Using Bands](#) for more information.
- Defining and editing regions, which are used to identify a group of values within the rectangular area. See the topic [Using Regions](#) for more information.
- Marking and unmarking elements to hand select the values that could be used to generate a Select or Derive node. See the topic [Using Marked Elements](#) for more information.
- Generating nodes using the values identified by bands, regions, marked elements, and web links to use in your stream. See the topic [Generating Nodes from Graphs](#) for more information.

Figure 1. Graph with the generate menu showing



- [Using Bands](#)
- [Using Regions](#)
- [Using Marked Elements](#)
- [Generating Nodes from Graphs](#)

Related information

- [Using Bands](#)
- [Using Regions](#)
- [Using Marked Elements](#)
- [Generating Nodes from Graphs](#)
- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Plot Node](#)
- [Distribution Node](#)
- [Histogram Node](#)
- [Collection Node](#)
- [Multiplot Node](#)
- [Web Node](#)
- [Time Plot Node](#)
- [Evaluation node](#)
- [Working with Graph Output](#)
- [Editing Visualizations](#)
- [Adding Titles and Footnotes](#)
- [Using Graph Stylesheets](#)
- [Printing, saving, copying, and exporting graphs](#)

Using Bands

In any graph with a scale field on the x axis, you can draw vertical band lines to split the range of values on the x axis. If a graph has multiple panels, a band line drawn in one panel is also represented in the other panels as well.

Not all graphs accept bands. Some of those graphs which can have bands include: histograms, bar charts and distributions, plots (line, scatter, time, etc.), collections, and evaluation charts. In graphs with paneling, bands appear in all panels. And in some cases in a SPLOM, you will see a horizontal band line since the axis on which the field/variable band was drawn has been flipped.

Figure 1. Graph with three bands

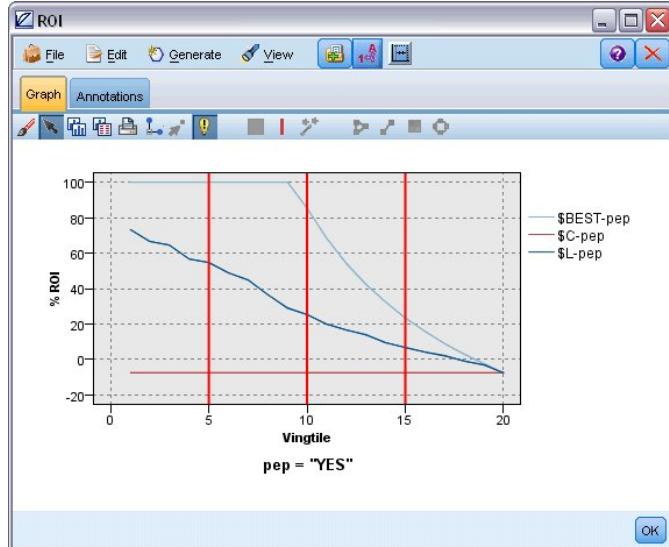
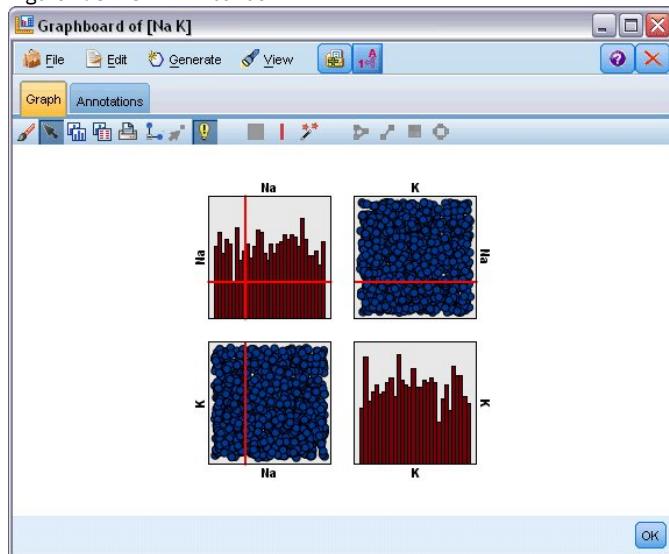


Figure 2. SPLOM with bands



Defining Bands

In a graph without bands, adding a band line splits the graph into two bands. The band line value represents the starting point, also referred to as the lower bound, of the second band when reading the graph from left to right. Likewise, in a graph with two bands, adding a band line splits one of those bands into two, which results in three bands. By default, bands are named *bandN*, where *N* equals the number of bands from left to right on the x axis.

Once you have defined a band, you can drag-and-drop the band to reposition it on the x axis. You can see more shortcuts by right-clicking within a band for tasks such as renaming, deleting, or generating nodes for that specific band.

To define bands:

1. Verify that you are in Explore mode. From the menus, choose View > Explore Mode.
2. In the Explore mode toolbar, click the Draw Band button.

Figure 3. Draw Bands toolbar button



3. In a graph that accepts bands, click the x-axis value point at which you want to define a band line.

Note: Alternatively, click the Split Graph into Bands toolbar icon and enter the number of equal bands you want and click Split.

Figure 4. Splitter icon used to expand the toolbar with options for splitting into bands



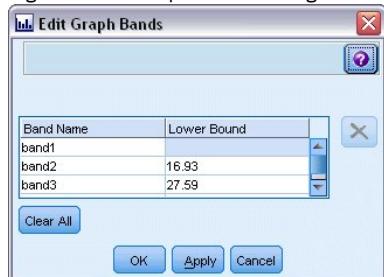
Figure 5. Creating equal bands toolbar with bands enabled



Editing, Renaming, and Deleting Bands

You can edit the properties of existing bands in the Edit Graph Bands dialog box or through context menus in the graph itself.

Figure 6. Edit Graph Bands dialog box



To edit bands:

1. Verify that you are in Explore mode. From the menus, choose View > Explore Mode.
2. In the Explore mode toolbar, click the Draw Band button.
3. From the menus, choose Edit > Graph Bands. The Edit Graph Bands dialog box opens.
4. If you have multiple fields in your graph (such as with SPLOM graphs), you can select the field you want in the drop-down list.
5. Add a new band by typing a name and lower bound. Press the Enter key to begin a new row.
6. Edit a band's boundary by adjusting the Lower Bound value.
7. Rename a band by entering a new band name.
8. Delete a band by selecting the line in the table and clicking the delete button.
9. Click OK to apply your changes and close the dialog box.

Note: Alternatively, you can delete and rename bands directly in the graph by right-clicking the band's line and choosing the option you want from the context menus.

Related information

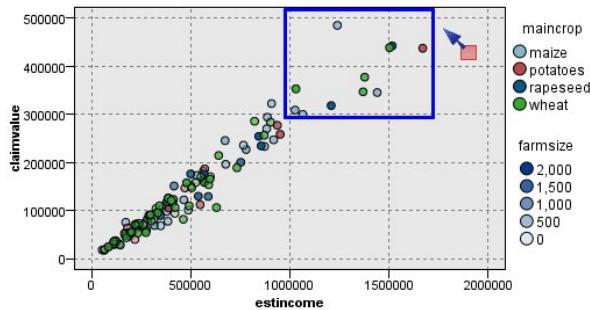
- [Exploring Graphs](#)
- [Using Regions](#)
- [Using Marked Elements](#)
- [Generating Nodes from Graphs](#)

Using Regions

In any graph with two scale (or range) axes, you can draw regions to group values within a rectangular area you draw, called a region. A **region** is an area of the graph described by its minimum and maximum X and Y values. If a graph has multiple panels, a region drawn in one panel is also represented in the other panels as well.

Not all graphs accept regions. Some of those graphs that accept regions include: plots (line, scatter, bubble, time, etc.), SPLOM, and collections. These regions are drawn in X,Y space and cannot, therefore, be defined in 1-D, 3-D, or animated plots. In graphs with paneling, regions appear in all panels. With a scatterplot matrix (SPLOM), a corresponding region will appear in the corresponding upper plots but not on the diagonal plots since they show only one scale field.

Figure 1. Defining a region of high claim values



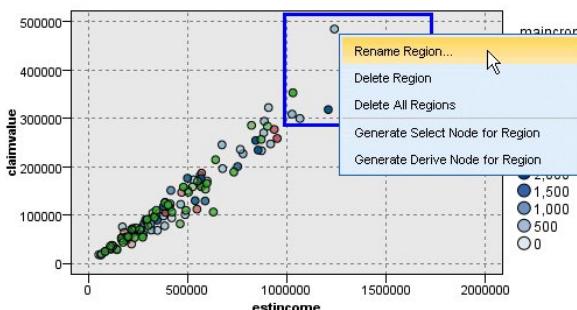
Defining Regions

Wherever you define a region, you are creating a grouping of values. By default, each new region is called *Region<N>*, where *N* corresponds to the number of regions already created.

Once you have a defined a region, you can right-click the region line to get some basic shortcuts. However, you can see many other shortcuts by right-clicking inside the region (not on the line) for tasks such as renaming, deleting, or generating Select and Derive nodes for that specific region.

You can select subsets of records on the basis of their inclusion in a particular region or in one of several regions. You can also incorporate region information for a record by producing a Derive node to flag records based on their inclusion in a region. See the topic [Generating Nodes from Graphs](#) for more information.

Figure 2. Exploring the region of high claim values



To define regions:

1. Verify that you are in Explore mode. From the menus, choose View > Explore Mode.
2. In the Explore mode toolbar, click the Draw Region button.

Figure 3. Draw Region toolbar button



3. In a graph that accepts regions, click and drag your mouse to draw the rectangular region.

Editing, Renaming and Deleting Regions

You can edit the properties of existing regions in the Edit Graph Regions dialog box or through context menus in the graph itself.

Figure 4. Specifying properties for the defined regions



To edit regions:

1. Verify that you are in Explore mode. From the menus, choose View > Explore Mode.
2. In the Explore mode toolbar, click the Draw Region button.
3. From the menus, choose Edit > Graph Regions. The Edit Graph Regions dialog box opens.
4. If you have multiple fields in your graph (for example, SPLOM graphs), you must define the field for the region in the *Field A* and *Field B* columns.
5. Add a new region on a new line by typing a name, selecting field names (if applicable) and defining the maximum and minimum boundaries for each field. Press the Enter key to begin a new row.
6. Edit existing region boundaries by adjusting the Min and Max values for *A* and *B*.
7. Rename a region by changing the region's name in the table.
8. Delete a region by selecting the line in the table and clicking the delete button.
9. Click OK to apply your changes and close the dialog box.

Note: Alternatively, you can delete and rename regions directly in the graph by right-clicking the region's line and choosing the option you want from the context menus.

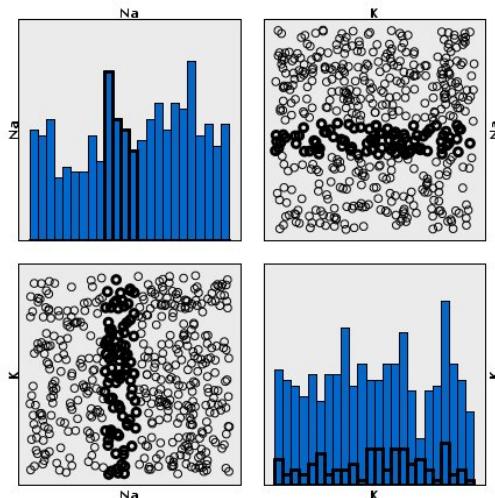
Related information

- [Exploring Graphs](#)
 - [Using Bands](#)
 - [Using Marked Elements](#)
 - [Generating Nodes from Graphs](#)
-

Using Marked Elements

You can mark elements, such as bars, slices, and points, in any graph. Lines, areas, and surfaces cannot be marked in graphs other than time plot, multiplot, and evaluation graphs since lines refers to fields in those cases. Whenever you mark an element, you are essentially highlighting all of the data represented by that element. In any graph where the same case is represented in more than one place (such as SPLOM), marking is synonymous with brushing. You can mark elements in graphs and even within bands and regions. Whenever you mark an element and then go back into Edit mode, the marking still remains visible.

Figure 1. Marking elements in a SPLOM



You can mark and unmark elements by clicking on elements in the graph. When you first click an element to mark it, the element appears with a thick border color to indicate that it has been marked. If you click the element again, the border disappears and the element is no longer marked.

To mark multiple elements, you can either hold down the Ctrl key while clicking elements, or you can drag the mouse around each of the elements you want marked using the "magic wand". Remember that if you click another area or element without holding down the Ctrl key, all previously marked elements are cleared.

You can generate Select and Derive nodes from the marked elements in your graph. See the topic [Generating Nodes from Graphs](#) for more information.

To mark elements:

1. Verify that you are in Explore mode. From the menus, choose View > Explore Mode.
2. In the Explore mode toolbar, click the Mark Elements button.

Figure 2. Mark Elements toolbar button



3. Either click on the element you require, or click and drag your mouse to draw a line around the region containing multiple elements.

Related information

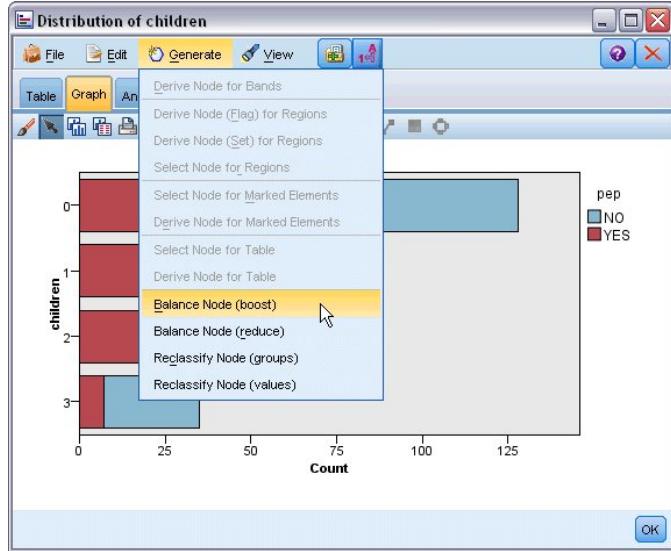
- [Exploring Graphs](#)
- [Using Bands](#)
- [Using Regions](#)
- [Generating Nodes from Graphs](#)

Generating Nodes from Graphs

One of the most powerful features offered by IBM® SPSS® Modeler graphs is the ability to generate nodes from a graph or a selection within the graph. For example, from a time plot graph, you can generate Derive and Select nodes based on a selection or region of data, effectively "subsetting" the data. For example, you might use this powerful feature to identify and exclude outliers.

Whenever you can draw a band, you can also generate a Derive node. In graphs with two scale axes, you can generate Derive or Select nodes from the regions drawn in your graph. In graphs with marked elements, you can generate Derive nodes, Select nodes, and in some cases Filter nodes from these elements. Balance node generation is enabled for any graph showing a distribution of counts.

Figure 1. Graph with the generate menu showing



Whenever you generate a node, it is placed on the stream canvas directly so that you can connect it to an existing stream. The following nodes can be generated from graphs: Select, Derive, Balance, Filter, and Reclassify.

Select Nodes

Select nodes can be generated to test for inclusion of the records within a region and exclusion of all records falling outside the region or the reverse for downstream processing.

- **For bands.** You can generate a Select node that includes or excludes the records within that band. Select node for Bands only is only available through contextual menus since you need to select which band to use in the Select node.
- **For regions.** You can generate a Select node that includes or excludes the records within a region.

- **For marked elements.** You can generate Select nodes to capture the records corresponding to the marked elements or web graph links.

Derive Nodes

Derive nodes can be generated from regions, bands, and marked elements. All graphs can produce Derive nodes. In the case of evaluation charts, a dialog box for selecting the model appears. In the case of web graphs, Derive Node ("And") and Derive Node ("Or") are possible.

- **For bands.** You can generate a Derive node that produces a category for each interval marked on the axis, using the band names listed in the Edit Bands dialog box as category names.
- **For regions.** You can generate a Derive node (Derive as flag) that creates a flag field called *in_region* with the flags set to *T* for records inside any region and *F* for records outside all regions. You can also generate a Derive node (Derive as set) that produces a set with a value for each region with a new field called *region* for each record, which takes as its value the name of the region into which the records fall. Records falling outside all regions receive the name of the default region. Value names become the region names listed in the Edit regions dialog box.
- **For marked elements.** You can generate a Derive node that calculates a flag that is *True* for all marked elements and *False* for all other records.

Balance Nodes

Balance nodes can be generated to correct imbalances in the data, such as reducing the frequency of common values (use Balance Node (reduce) menu option) or boosting the occurrence of infrequent values (use Balance Node (boost) menu option). Balance node generation is enabled for any graph showing a distribution of counts, such as Histogram, Dot, Collection, Bar of Counts, Pie of Counts, and Multiplot.

Filter Nodes

Filter nodes can be generated to rename or filter fields based on the lines or nodes marked in the graph. In the case of evaluation charts, the best fit line does not generate a filter node.

Reclassify Nodes

Reclassify nodes can be generated to recode values. This option is used for distribution graphs. You can generate a Reclassify node for **groups** to recode specific values of a displayed field depending upon their inclusion in a group (select groups using Ctrl+click on the **Tables** tab). You can also generate a reclassify node for **values** to recode data into an existing set of numerous values, such as reclassifying data into a standard set of values in order to merge financial data from various companies for analysis.

Note: If the values are predefined, you can read them into IBM SPSS Modeler as a flat file and use a distribution to display all values. Then generate a Reclassify (values) node for this field directly from the chart. Doing so will put all the target values in the Reclassify node's *New values* column (drop-down list).

When setting options for the Reclassify node, the table enables a clear mapping from old set values to new values you specify:

- **Original value.** This column lists existing values for the select field(s).
- **New value.** Use this column to type new category values or select one from the drop-down list. When you automatically generate a Reclassify node using values from a Distribution chart, these values are included in the drop-down list. This allows you to quickly map existing values to a known set of values. For example, healthcare organizations sometimes group diagnoses differently based upon network or locale. After a merger or acquisition, all parties will be required to reclassify new or even existing data in a consistent fashion. Rather than manually typing each target value from a lengthy list, you can read the master list of values in to IBM SPSS Modeler, run a Distribution chart for the *Diagnosis* field, and generate a Reclassify (values) node for this field directly from the chart. This process will make all of the target Diagnosis values available from the New Values drop-down list.

For more information about the Reclassify Node, see [Setting Options for the Reclassify Node](#).

Generating Nodes from Graphs

You can use the Generate menu in the graph output window to generate nodes. The generated node will be placed on the stream canvas. To use the node, connect it to an existing stream.

To generate a node from a graph:

1. Verify that you are in Explore mode. From the menus, choose View > Explore Mode.
2. In the Explore mode toolbar, click the Region button.
3. Define bands, regions, or any marked elements needed to generate your node.
4. From the Generate menu, choose the kind of node you want to produce. Only those which are possible are enabled.

Note: Alternatively, you can also generate nodes directly from the graph by right-clicking and choosing the generate option you want from the context menus.

Related information

- [Exploring Graphs](#)
- [Using Bands](#)
- [Using Regions](#)
- [Using Marked Elements](#)

Editing Visualizations

While Explore mode allows you to analytically explore the data and values represented by the visualization, Edit mode allows you to change the visualization's layout and look. For example, you can change the fonts and colors to match your organization's style guide. To select this mode, choose View > Edit Mode from the menus (or click the toolbar icon).

In Edit mode, there are several toolbars that affect different aspects of the visualization's layout. If you find that there are any you don't use, you can hide them to increase the amount of space in the dialog box in which the graph is displayed. To select or deselect toolbars, click the relevant toolbar name on the View menu.

Note: To add further detail to your visualizations, you can apply title, footnote, and axis labels. See the topic [Adding Titles and Footnotes](#) for more information.

You have several options for editing a visualization in **Edit mode**. You can:

- Edit text and format it.
- Change the fill color, transparency, and pattern of frames and graphic elements.
- Change the color and dashing of borders and lines.
- Rotate and change the shape and aspect ratio of point elements.
- Change the size of graphic elements (such as bars and points).
- Adjust the space around items by using margins and padding.
- Specify formatting for numbers.
- Change the axis and scale settings.
- Sort, exclude, and collapse categories on a categorical axis.
- Set the orientation of panels.
- Apply transformations to a coordinate system.
- Change statistics, graphic element types, and collision modifiers.
- Change the position of the legend.
- Apply visualization stylesheets.

The following topics describe how to perform these various tasks. It is also recommended that you read the general rules for editing graphs.

How to Switch to Edit Mode

From the menus choose:

[View > Edit Mode](#)

- [General Rules for Editing Visualizations](#)
- [Editing and Formatting Text](#)
- [Changing Colors, Patterns, Dashings, and Transparency](#)
- [Rotating and Changing the Shape and Aspect Ratio of Point Elements](#)
- [Changing the size of graphic elements](#)
- [Specifying Margins and Padding](#)
- [Formatting Numbers](#)
- [Changing the Axis and Scale Settings](#)
- [Editing Categories](#)
- [Changing the Orientation Panels](#)
- [Transforming the Coordinate System](#)
- [Changing Statistics and Graphic Elements](#)
- [Changing the Position of the Legend](#)
- [Copying a Visualization and Visualization Data](#)
- [Graphboard Editor Keyboard Shortcuts](#)
- [Adding Titles and Footnotes](#)
- [Using Graph Stylesheets](#)
- [Printing, saving, copying, and exporting graphs](#)

Related information

- [General Rules for Editing Visualizations](#)
- [Editing and Formatting Text](#)
- [Changing Colors, Patterns, Dashings, and Transparency](#)
- [Rotating and Changing the Shape and Aspect Ratio of Point Elements](#)
- [Changing the size of graphic elements](#)
- [Specifying Margins and Padding](#)
- [Formatting Numbers](#)
- [Changing the Axis and Scale Settings](#)
- [Editing Categories](#)

- [Changing the Orientation Panels](#)
- [Changing Statistics and Graphic Elements](#)
- [Changing the Position of the Legend](#)
- [Applying stylesheets](#)
- [Copying a Visualization and Visualization Data](#)
- [Graphboard Editor Keyboard Shortcuts](#)
- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Plot Node](#)
- [Distribution Node](#)
- [Histogram Node](#)
- [Collection Node](#)
- [Multiplot Node](#)
- [Web Node](#)
- [Time Plot Node](#)
- [Evaluation node](#)
- [Exploring Graphs](#)
- [Adding Titles and Footnotes](#)
- [Using Graph Stylesheets](#)
- [Printing, saving, copying, and exporting graphs](#)

General Rules for Editing Visualizations

Edit Mode

All edits are done in Edit mode. To enable Edit mode, from the menus choose:

View > Edit Mode

Selection

The options available for editing depend on selection. Different toolbar and properties palette options are enabled depending on what is selected. Only the enabled items apply to the current selection. For example, if an axis is selected, the Scale, Major Ticks, and Minor Ticks tabs are available in the properties palette.

Here are some tips for selecting items in the visualization:

- Click an item to select it.
- Select a graphic element (such as points in a scatterplot or bars in a bar chart) with a single click. After initial selection, click again to narrow the selection to groups of graphic elements or a single graphic element.
- Press Esc to deselect everything.

Palettes

When an item is selected in the visualization, the various palettes are updated to reflect the selection. The palettes contain controls for making edits to the selection. Palettes may be toolbars or a panel with multiple controls and tabs. Palettes can be hidden, so ensure the necessary palette is displayed for making edits. Check the View menu for palettes that are currently displayed.

You can reposition the palettes by clicking and dragging the empty space in a toolbar palette or the left side of other palettes. Visual feedback lets you know where you can dock the palette. For non-toolbar palettes, you can also click the close button to hide the palette and the undock button to display the palette in a separate window. Click the help button to display help for the specific palette.

Automatic Settings

Some settings provide an -auto- option. This indicates that automatic values are applied. Which automatic settings are used depends on the specific visualization and data values. You can enter a value to override the automatic setting. If you want to restore the automatic setting, delete the current value and press Enter. The setting will display -auto- again.

Removing/Hiding Items

You can remove/hide various items in the visualization. For example, you can hide the legend or axis label. To delete an item, select it and press Delete. If the item does not allow deletion, nothing will happen. If you accidentally delete an item, press Ctrl+Z to undo the deletion.

State

Some toolbars reflect the state of the current selection, others don't. The properties palette always reflects state. If a toolbar does not reflect state, this is mentioned in the topic that describes the toolbar.

Related information

- [Editing Visualizations](#)
-

Editing and Formatting Text

You can edit text in place and change the formatting of an entire text block. Note that you can't edit text that is linked directly to data values. For example, you can't edit a tick label because the content of the label is derived from the underlying data. However, you can format any text in the visualization.

How to Edit Text in Place

1. Double-click the text block. This action selects all the text. All toolbars are disabled at this time, because you cannot change any other part of the visualization while editing text.
2. Type to replace the existing text. You can also click the text again to display a cursor. Position the cursor where you want and enter the additional text.

How to Format Text

1. Select the frame containing the text. Do not double-click the text.
2. Format text using the font toolbar. If the toolbar is not enabled, make sure only the *frame* containing the text is selected. If the text itself is selected, the toolbar will be disabled.

You can change the font:

- Color
- Family (for example, Arial or Verdana)
- Size (the unit is pt unless you indicate a different unit, such as pc)
- Weight
- Alignment relative to the text frame

Formatting applies to all the text in a frame. You can't change the formatting of individual letters or words in any particular block of text.

Related information

- [Editing Visualizations](#)
 - [General Rules for Editing Visualizations](#)
-

Changing Colors, Patterns, Dashings, and Transparency

Many different items in a visualization have a fill and border. The most obvious example is a bar in a bar chart. The color of the bars is the fill color. They may also have a solid, black border around them.

There are other less obvious items in the visualization that have fill colors. If the fill color is transparent, you may not know there is a fill. For example, consider the text in an axis label. It appears as if this text is "floating" text, but it actually appears in a frame that has a transparent fill color. You can see the frame by selecting the axis label.

Any frame in the visualization can have a fill and border style, including the frame around the whole visualization. Also, any fill has an associated opacity/transparency level that can be adjusted.

How to Change the Colors, Patterns, Dashing, and Transparency

1. Select the item you want to format. For example, select the bars in a bar chart or a frame containing text. If the visualization is split by a categorical variable or field, you can also select the group that corresponds to an individual category. This allows you to change the default aesthetic assigned to that group. For example, you can change the color of one of the stacking groups in a stacked bar chart.
2. To change the fill color, the border color, or the fill pattern, use the color toolbar.

Note: This toolbar does not reflect the state of the current selection.

To change a color or fill, you can click the button to select the displayed option or click the drop-down arrow to choose another option. For colors, notice there is one color that looks like white with a red, diagonal line through it. This is the transparent color. You could use this, for example, to hide the borders on bars in a histogram.

- The first button controls the fill color. If the color is associated with a continuous or ordinal field, this button changes the fill color for the color associated with the highest value in the data. You can use the Color tab on the properties palette to change the color associated with the lowest value and missing data. The color of the elements will change incrementally from the Low color to the High color as the values of the underlying data increase.
- The second button controls the border color.

- The third button controls the fill pattern. The fill pattern uses the border color. Therefore, the fill pattern is visible only if there is a visible border color.
 - The fourth control is a slider and text box that control the opacity of the fill color and pattern. A lower percentage means less opacity and more transparency. 100% is fully opaque (no transparency).
3. To change the dashing of a border or line, use the line toolbar.
Note: This toolbar does not reflect the state of the current selection.

As with the other toolbar, you can click the button to select the displayed option or click the drop-down arrow to choose another option.

Related information

- [Editing Visualizations](#)
- [General Rules for Editing Visualizations](#)

Rotating and Changing the Shape and Aspect Ratio of Point Elements

You can rotate point elements, assign a different predefined shape, or change the aspect ratio (the ratio of width to height).

How to Modify Point Elements

1. Select the point elements. You cannot rotate or change the shape and aspect ratio of individual point elements.
2. Use the symbol toolbar to modify the points.
 - The first button allows you to change the shape of the points. Click the drop-down arrow and select a predefined shape.
 - The second button allows you to rotate the points to a specific compass position. Click the drop-down arrow and then drag the needle to the position you want.
 - The third button allows you to change the aspect ratio. Click the drop-down arrow and then click and drag the rectangle that appears. The shape of the rectangle represents the aspect ratio.

Related information

- [Editing Visualizations](#)
- [General Rules for Editing Visualizations](#)

Changing the size of graphic elements

You can change the size of the graphic elements in the visualization. These include bars, lines, and points among others. If the graphic element is sized by a variable or field, the specified size is the *minimum* size.

How to change the size of the graphic elements

1. Select the graphic elements you want to resize.
2. Use the slider to change the size.

Specifying Margins and Padding

If there is too much or too little spacing around or inside a frame in the visualization, you can change its margin and padding settings. The **margin** is the amount of space between the frame and other items around it. The **padding** is the amount of space between the border of the frame and the *contents* of the frame.

How to Specify Margins and Padding

1. Select the frame for which you want to specify margins and padding. This can be a text frame, the frame around the legend, or even the data frame displaying the graphic elements (such as bars and points).
2. Use the Margins tab on the properties palette to specify the settings. All sizes are in pixels unless you indicate a different unit (such as cm or in).

Related information

- [Editing Visualizations](#)

- [General Rules for Editing Visualizations](#)
-

Formatting Numbers

You can specify the format for numbers in tick labels on a continuous axis or data value labels displaying a number. For example, you can specify that numbers displayed in the tick labels are shown in thousands.

How to Specify Number Formats

1. Select the continuous axis tick labels or the data value labels if they contain numbers.
2. Click the Format tab on the properties palette.
3. Select the number formatting options you want:

Prefix. A character to display at the beginning of the number. For example, enter a dollar sign (\$) if the numbers are salaries in U.S. dollars.

Suffix. A character to display at the end of the number. For example, enter a percentage sign (%) if the numbers are percentages.

Min. Integer Digits. Minimum number of digits to display in the integer part of a decimal representation. If the actual value does not contain the minimum number of digits, the integer part of the value will be padded with zeros.

Max. Integer Digits. Maximum number of digits to display in the integer part of a decimal representation. If the actual value exceeds the maximum number of digits, the integer part of the value will be replaced with asterisks.

Min. Decimal Digits. Minimum number of digits to display in the decimal part of a decimal or scientific representation. If the actual value does not contain the minimum number of digits, the decimal part of the value will be padded with zeros.

Max. Decimal Digits. Maximum number of digits to display in the decimal part of a decimal or scientific representation. If the actual value exceeds the maximum number of digits, the decimal is rounded to the appropriate number of digits.

Scientific. Whether to display numbers in scientific notation. Scientific notation is useful for very large or very small numbers. -auto- lets the application determine when scientific notation is appropriate.

Scaling. A scale factor, which is a number by which the original value is divided. Use a scale factor when the numbers are large, but you don't want the label to extend too much to accommodate the number. If you change the number format of the tick labels, be sure to edit the axis title to indicate how the number should be interpreted. For example, assume your scale axis displays salaries and the labels are 30,000, 50,000, and 70,000. You might enter a scale factor of 1000 to display 30, 50, and 70. You should then edit the scale axis title to include the text in thousands.

Parentheses for -ve. Whether parentheses should be displayed around negative values.

Grouping. Whether to display a character between groups of digits. Your computer's current locale determines which character is used for digit grouping.

Related information

- [Editing Visualizations](#)
 - [General Rules for Editing Visualizations](#)
-

Changing the Axis and Scale Settings

There are several options for modifying axes and scales.

How to change axis and scale settings

1. Select any part of the axis (for example, the axis label or tick labels).
2. Use the Scale, Major Ticks, and Minor Ticks tabs on the properties palette to change the axis and scale settings.

Scale tab

Note: The Scale tab does not appear for graphs where the data is pre-aggregated (for example, histograms).

Type. Specifies whether the scale is linear or transformed. Scale transformations help you understand the data or make assumptions necessary for statistical inference. On scatterplots, you might use a transformed scale if the relationship between the independent and dependent variables or fields is nonlinear. Scale transformations can also be used to make a skewed histogram more symmetric so that it resembles a normal distribution. Note that you are transforming only the scale on which the data are displayed; you are not transforming the actual data.

- **linear.** Specifies a linear, untransformed scale.

- **log.** Specifies a base-10 log transformed scale. To accommodate zero and negative values, this transformation uses a modified version of the log function. This "safe log" function is defined as `sign(x)`
`* log(1 + abs(x))`. So `safeLog(-99)` equals:
`sign(-99)`
`* log(1 + abs(-99)) = -1 * log(1 + 99) = -1 * 2 = -2`
- **power.** Specifies a power transformed scale, using an exponent of 0.5. To accommodate negative values, this transformation uses a modified version of the power function. This "safe power" function is defined as `sign(x) * pow(abs(x), 0.5)`. So `safePower(-100)` equals:
`sign(-100)`
`* pow(abs(-100), 0.5) = -1 * pow(100, 0.5) = -1 * 10 = -10`

Min/Max/Nice Low/Nice High. Specifies the range for the scale. Selecting Nice Low and Nice High allows the application to select an appropriate scale based on the data. The minimum and maximum are "nice" because they are typically whole values greater or less than the maximum and minimum data values. For example, if the data range from 4 to 92, a nice low and high for scale may be 0 and 100 rather than the actual data minimum and maximum. Be careful that you don't set a range that is too small and hides important items. Also note that you cannot set an explicit minimum and maximum if the Include zero option is selected.

Low Margin/High Margin. Create margins at the low and/or high end of the axis. The margin appears perpendicular to the selected axis. The unit is pixels unless you indicate a different unit (such as cm or in). For example, if you set the High Margin to 5 for the vertical axis, a horizontal margin of 5 px runs along the top of the data frame.

Reverse. Specifies whether the scale is reversed.

Include zero. Indicates that the scale should include 0. This option is commonly used for bar charts to ensure the bars begin at 0, rather than a value near the height of the smallest bar. If this option is selected, Min and Max are disabled because you cannot set a custom minimum and maximum for the scale range.

Major Ticks/Minor Ticks Tabs

Ticks or tick marks are the lines that appear on an axis. These indicate values at specific intervals or categories. **Major ticks** are the tick marks with labels. These are also longer than other tick marks. **Minor ticks** are tick marks that appear between the major tick marks. Some options are specific to the tick type, but most options are available for major and minor ticks.

Show ticks. Specifies whether major or minor ticks are displayed on a graph.

Show gridlines. Specifies whether gridlines are displayed at the major or minor ticks. **Gridlines** are lines that cross a whole graph from axis to axis.

Position. Specifies the position of the tick marks relative to the axis.

Length. Specifies the length of the tick marks. The unit is pixels unless you indicate a different unit (such as cm or in).

Base. Applies only to major ticks. Specifies the value at which the first major tick appears.

Delta. Applies only to major ticks. Specifies the difference between major ticks. That is, major ticks will appear at every n th value, where n is the delta value.

Divisions. Applies only to minor ticks. Specifies the number of minor tick divisions between major ticks. The number of minor ticks is one less than the number of divisions. For example, assume that there are major ticks at 0 and 100. If you enter 2 as the number of minor tick divisions, there will be one minor tick at 50, dividing the 0–100 range and creating two divisions.

Related information

- [Editing Visualizations](#)
 - [General Rules for Editing Visualizations](#)
-

Editing Categories

You can edit the categories on a categorical axis in several ways:

- Change the sort order for displaying the categories.
- Exclude specific categories.
- Add a category that does not appear in the data set.
- Collapse/combine small categories into one category.

How to Change the Sort Order of Categories

1. Select a categorical axis. The Categories palette displays the categories on the axis.

Note: If the palette is not visible, make sure that you have it enabled. From the View menu in IBM® SPSS® Modeler, choose Categories.

2. In the Categories palette, select a sorting option from the drop-down list:

Custom. Sort categories based on the order in which they appear in the palette. Use the arrow buttons to move categories to the top of the list, up, down, and to the bottom of the list.

Data. Sort categories based on the order in which they occur in the dataset.

Name. Sort categories alphabetically, using the names as displayed in the palette. This may be either the value or label, depending on whether the toolbar button to display values and labels is selected.

Value. Sort categories by the underlying data value, using the values displayed in parentheses in the palette. Only data sources with metadata (such as IBM SPSS Statistics data files) support this option.

Statistic. Sort categories based on the calculated statistic for each category. Examples of statistics include counts, percentages, and means. This option is available only if a statistic is used in the graph.

How to Add a Category

By default, only categories that appear in the data set are available. You can add a category to the visualization if needed.

1. Select a categorical axis. The Categories palette displays the categories on the axis.

Note: If the palette is not visible, make sure that you have it enabled. From the View menu in IBM SPSS Modeler, choose Categories.

2. In the Categories palette, click the add category button:

Figure 1. Add category button



3. In the Add a new category dialog box, enter a name for the category.

4. Click OK.

How to Exclude Specific Categories

1. Select a categorical axis. The Categories palette displays the categories on the axis.

Note: If the palette is not visible, make sure that you have it enabled. From the View menu in IBM SPSS Modeler, choose Categories.

2. In the Categories palette, select a category name in the Include list, and then click the X button. To move the category back, select its name in the Excluded list, and then click the arrow to the right of the list.

How to Collapse/Combine Small Categories

You can combine categories that are so small you don't need to display them separately. For example, if you have a pie chart with many categories, consider collapsing categories with a percentage less than 10. Collapsing is available only for statistics that are additive. For example, you can't add means together because means are not additive. Therefore, combining/collapsing categories using a mean is not available.

1. Select a categorical axis. The Categories palette displays the categories on the axis.

Note: If the palette is not visible, make sure that you have it enabled. From the View menu in IBM SPSS Modeler, choose Categories.

2. In the Categories palette, select Collapse and specify a percentage. Any categories whose percentage of the total is less than the specified number are combined into one category. The percentage is based on the statistic shown in the chart. Collapsing is available only for count-based and summation (sum) statistics.

Related information

- [Editing Visualizations](#)
 - [General Rules for Editing Visualizations](#)
-

Changing the Orientation Panels

If you are using panels in your visualization, you can change their orientation.

How to Change the Orientation of the Panels

1. Select any part of the visualization.
2. Click the Panels tab on the properties palette.
3. Select an option from Layout:

Table. Lays out panels like a table, in that there is a row or column assigned to every individual value.

Transposed. Lays out panels like a table, but also swaps the original rows and columns. This option is not the same as transposing the graph itself. Note that the x axis and the y axis are unchanged when you select this option.

List. Lays out panels like a list, in that each cell represents a combination of values. Columns and rows are no longer assigned to individual values. This option allows the panels to wrap if needed.

Related information

- [Editing Visualizations](#)
 - [General Rules for Editing Visualizations](#)
-

Transforming the Coordinate System

Many visualizations are displayed in a flat, rectangular coordinate system. You can transform the coordinate system as needed. For example, you can apply a polar transformation to the coordinate system, add oblique drop shadow effects, and transpose the axes. You can also undo any of these transformations if they are already applied to the current visualization. For example, a pie chart is drawn in a polar coordinate system. You can undo the polar transformation and display the pie chart as a single stacked bar in a rectangular coordinate system.

How to Transform the Coordinate System

1. Select the coordinate system that you want to transform. You select the coordinate system by selecting the frame around the individual graph.
2. Click the Coordinates tab on the properties palette.
3. Select the transformations that you want to apply to the coordinate system. You can also deselect a transformation to undo it.
Transposed. Changing the orientation of the axes is called **transposing**. It is similar to swapping the vertical and horizontal axes in a 2-D visualization.

Polar. A polar transformation draws the graphic elements at a specific angle and distance from the center of the graph. A pie chart is a 1-D visualization with a polar transformation that draws the individual bars at specific angles. A radar chart is a 2-D visualization with a polar transformation that draws graphic elements at specific angles and distances from the center of the graph. A 3-D visualization would also include an additional depth dimension.

Oblique. An oblique transformation adds a 3-D effect to the graphic elements. This transformation adds depth to the graphic elements, but the depth is purely decorative. It is not influenced by particular data values.

Same Ratio. Applying the same ratio specifies that the same distance on each scale represents the same difference in data values. For example, 2cm on both scales represent a difference of 1000.

Pre-transform inset %. If axes are clipped after the transformation, you may want to add insets to the graph before applying the transformation. The insets shrink the dimensions by a certain percentage before any transformations are applied to the coordinate system. You have control over the lower x, upper x, lower y, and upper y dimensions, in that order.

Post-transform inset %. If you want to change the aspect ratio of the graph, you can add insets to the graph after applying the transformation. The insets shrink the dimensions by a certain percentage after any transformations are applied to the coordinate system. These insets can also be applied even if no transformation is applied to the graph. You have control over the lower x, upper x, lower y, and upper y dimensions, in that order.

Related information

- [Editing Visualizations](#)
 - [General Rules for Editing Visualizations](#)
-

Changing Statistics and Graphic Elements

You can convert a graphic element to another type, change the statistic used to draw the graphic element, or specify the collision modifier that determines what happens when graphic elements overlap.

How to Convert a Graphic Element

1. Select the graphic element that you want to convert.
2. Click the Element tab on the properties palette.
3. Select a new graphic element type from the Type list.

Table 1. Graphic element types

Graphic Element Type	Description
Point	A marker identifying a specific data point. A point element is used in scatterplots and other related visualizations.
Interval	A rectangular shape drawn at a specific data value and filling the space between an origin and another data value. An interval element is used in bar charts and histograms.
Line	A line that connects data values.
Path	A line that connects data values in the order they appear in the dataset.
Area	A line that connects data elements with the area between the line and an origin filled in.
Polygon	A multi-sided shape enclosing a data region. A polygon element could be used in a binned scatterplot or a map.
Schema	An element consisting of a box with whiskers and markers indicating outliers. A schema element is used for boxplots.

How to Change the Statistic

1. Select the graphic element whose statistic you want to change.
2. Click the Element tab on the properties palette.

How to Specify the Collision Modifier

The collision modifier determines what happens when graphic elements overlap.

1. Select the graphic element for which you want to specify the collision modifier.
2. Click the Element tab on the properties palette.
3. From the Modifier drop-down list, select a collision modifier. -auto- lets the application determine which collision modifier is appropriate for the graphic element type and statistic.

Overlay. Draw graphic elements on top of each other when they have the same value.

Stack. Stack graphic elements that would normally be superimposed when they have the same data values.

Dodge. Move graphic elements next to other graphic elements that appear at the same value, rather than superimposing them. The graphic elements are arranged symmetrically. That is, the graphic elements are moved to opposite sides of a central position. Dodging is very similar to clustering.

Pile. Move graphic elements next to other graphic elements that appear at the same value, rather than superimposing them. The graphic elements are arranged asymmetrically. That is, the graphic elements are piled on top of one another, with the graphic element on the bottom positioned at a specific value on the scale.

Jitter (normal). Randomly reposition graphic elements at the same data value using a normal distribution.

Jitter (uniform). Randomly reposition graphic elements at the same data value using a uniform distribution.

Related information

- [Editing Visualizations](#)
 - [General Rules for Editing Visualizations](#)
-

Changing the Position of the Legend

If a graph includes a legend, the legend is typically displayed to the right of a graph. You can change this position if needed.

How to Change the Legend Position

1. Select the legend.
2. Click the Legend tab on the properties palette.
3. Select a position.

Related information

- [Editing Visualizations](#)
 - [General Rules for Editing Visualizations](#)
-

Copying a Visualization and Visualization Data

The General palette includes buttons for copying the visualization and its data.

Figure 1. Copy visualization button



Copying the visualization. This action copies the visualization to the clipboard as an image. Multiple image formats are available. When you paste the image into another application, you can choose a "paste special" option to select one of the available image formats for pasting.

Figure 2. Copy visualization data button



Copying the visualization data. This action copies the underlying data that is used to draw the visualization. The data is copied to the clipboard as plain text or HTML-formatted text. When you paste the data into another application, you can choose a "paste special" option to choose one of these formats for pasting.

Related information

- [Editing Visualizations](#)
 - [General Rules for Editing Visualizations](#)
-

Graphboard Editor Keyboard Shortcuts

Table 1. Keyboard shortcuts

Shortcut Key	Function
Ctrl+Space	Toggle between Explore and Edit mode
Delete	Delete a visualization item
Ctrl+Z	Undo
Ctrl+Y	Redo
F2	Display outline for selecting items in the graph

Related information

- [Editing Visualizations](#)
-

Adding Titles and Footnotes

For all graph types you can add unique title, footnote, or axis labels to help identify what is shown in the graph.

Adding Titles to Graphs

1. From the menus, choose Edit > Add Graph Title. A text box containing <TITLE> is displayed above the graph.
2. Verify that you are in Edit mode. From the menus, choose View > Edit Mode.
3. Double-click the <TITLE> text.
4. Type the required title and press Return.

Adding Footnotes to Graphs

1. From the menus, choose Edit > Add Graph Footnote. A text box containing <FOOTNOTE> is displayed below the graph.
2. Verify that you are in Edit mode. From the menus, choose View > Edit Mode.
3. Double-click the <FOOTNOTE> text.
4. Type the required title and press Return.

Related information

- [Using Graph Stylesheets](#)
- [Printing, saving, copying, and exporting graphs](#)
- [Common Graph Nodes Features](#)
- [Graphboard node](#)
- [Plot Node](#)

- [Distribution Node](#)
 - [Histogram Node](#)
 - [Collection Node](#)
 - [Multiplot Node](#)
 - [Web Node](#)
 - [Time Plot Node](#)
 - [Evaluation node](#)
 - [Working with Graph Output](#)
 - [Exploring Graphs](#)
 - [Editing Visualizations](#)
-

Using Graph Stylesheets

Basic graph display information, such as colors, fonts, symbols, and line thickness, are controlled by a stylesheet. There is a default stylesheet supplied with IBM® SPSS® Modeler; however, you can make changes to it if you need. For example, you may have a corporate color scheme for presentations that you want used in your graphs. See the topic [Editing Visualizations](#) for more information.

In the graph nodes, you can use the Edit mode to make style changes to the look of a graph. You can then use the Edit > Styles menu to save the changes as a stylesheet to apply to all graphs that you subsequently generate from the current graph node or as a new default stylesheet for all graphs that you produce using IBM SPSS Modeler.

There are five stylesheet options available from the Styles option on the Edit menu:

- **Switch Stylesheet.** This displays a list of different, stored, stylesheets that you can select to change the look of your graphs. See the topic [Applying stylesheets](#) for more information.
- **Store Styles in Node.** This stores modifications to the selected graph's styles so that they are applied to any future graphs created from the same graph node in the current stream.
- **Store Styles as Default.** This stores modifications to the selected graph's styles so that they are applied to all future graphs created from any graph node in any stream. After selecting this option, you can use Apply Default Styles to change any other existing graphs to use the same styles.
- **Apply Default Styles.** This changes the selected graph's styles to those that are currently saved as the default styles.
- **Apply Original Styles.** This changes a graph's styles back to the ones supplied as the original default.
- [Applying stylesheets](#)

Related information

- [Adding Titles and Footnotes](#)
 - [Printing, saving, copying, and exporting graphs](#)
 - [Common Graph Nodes Features](#)
 - [Graphboard node](#)
 - [Plot Node](#)
 - [Distribution Node](#)
 - [Histogram Node](#)
 - [Collection Node](#)
 - [Multiplot Node](#)
 - [Web Node](#)
 - [Time Plot Node](#)
 - [Evaluation node](#)
 - [Working with Graph Output](#)
 - [Exploring Graphs](#)
 - [Editing Visualizations](#)
-

Applying stylesheets

You can apply a visualization stylesheet that specifies stylistic properties of the visualization. For example, the stylesheet can define fonts, dashings, and colors, among other options. To a certain extent, stylesheets provide a shortcut for edits that you would have to perform manually. Note, however, that a stylesheet is limited to *style* changes. Other changes such as the position of the legend or the scale range are not stored in the stylesheet.

How to Apply a Stylesheet

1. From the menus choose:
Edit > Styles > Switch Stylesheet

2. Use the Switch Stylesheet dialog box to select a stylesheet.
3. Click Apply to apply the stylesheet to the visualization without closing the dialog. Click OK to apply the stylesheet and close the dialog box.

Switch>Select Stylesheet Dialog Box

The table at the top of the dialog box lists all of the visualization stylesheets that are currently available. Some stylesheets are pre-installed, while others may have been created in the IBM® SPSS® Visualization Designer (a separate product).

The bottom of the dialog box shows example visualizations with sample data. Select one of the stylesheets to apply its styles to the example visualizations. These examples can help you determine how the stylesheet will affect your actual visualization.

The dialog box also offers the following options.

Existing styles. By default, a stylesheet can overwrite all the styles in the visualization. You can change this behavior.

- **Overwrite all styles.** When applying the stylesheet, overwrite all styles in the visualization, including those styles modified in the visualization during the current editing session.
- **Preserve modified styles.** When applying the stylesheet, overwrite only those styles that were *not* modified in the visualization during the current editing session. Styles that were modified during the current editing session are preserved.

Manage. Manage visualization templates, stylesheets, and maps on your computer. You can import, export, rename, and delete visualization templates, stylesheets, and maps on your local machine. See the topic [Managing Templates, Stylesheets, and Map Files](#) for more information.

Location. Change the location in which visualization templates, stylesheets, and maps are stored. The current location is listed to the right of the button. See the topic [Setting the Location of Templates, Stylesheets, and Maps](#) for more information.

Printing, saving, copying, and exporting graphs

Each graph has a number of options that enable you to save or print the graph or export it to another format. Most of these options are available from the File menu. In addition, from the Edit menu, you can choose to copy the graph, the data within it, or the Microsoft Office Drawing Object for use in another application.

Printing

To print the graph, use the Print menu item or button. Before you print, you can use Page Setup and Print Preview to set print options and preview the output. For more information about configuring page headers and footers, see [Setting Header and Footer Preferences](#).

Saving graphs

To save the graph to an IBM® SPSS® Modeler output file (*.cou), choose File > Save or File > Save As from the menus.

or

To save the graph in the repository, choose File > Store Output from the menus.

Copying graphs

To copy the graph for use in another application, such as MS Word or MS PowerPoint, choose Edit > Copy Graph from the menus.

Copying data

To copy the data for use in another application, such as MS Excel or MS Word, choose Edit > Copy Data from the menus. By default, the data will be formatted as HTML. Use Paste Special in the other application to view other formatting options when pasting.

Copying Microsoft Office Graphic Object

You can copy a graph as a Microsoft Office Graphic Object and use it in Microsoft Office applications such as Excel or PowerPoint. To copy a graph, choose Edit > Copy Microsoft Office Graphic Object from the menus. The content will be copied to your clipboard, and will be in binary format by default. Use Paste Special in the Microsoft Office application to specify other formatting options when pasting.

Note that some content may not support this feature, in which case the Copy Microsoft Office Graphic Object menu option will be disabled. Also note that the appearance of the graph may be different after pasting into an Office application, but the graph data is the same.

There are six types of graph output that can be copied and pasted into Excel: Simple Bar, Stacked Bar, Simple Boxplot, Clustered Boxplot, Simple Scatter, and Grouped Scatter. If using Panel and Animation options for any of these graph types, the Copy Microsoft Office Graphic Object option

will be disabled in SPSS Modeler. And for other settings such as Optional Aesthetics or Overlay, the option is partially supported. See the following table for details:

Table 1. Copy Microsoft Graphic Object support

Graph Output Template	Modeler Graph Node	Modeler Graph Type	Basic Setting	Optional Aesthetics	Overlay	Microsoft Graphic Object Copy Support	Comments
Simple Bar	Graphboard	Bar	Yes	No	N/A	Yes	
		Bar of Counts	Yes	No	N/A	Yes	
		Distribution	Bar	N/A	No	Yes	
Stacked Bar	Graphboard	Bar	Yes	Yes	N/A	Yes with limitation	Only Yes for the categorical variable in Optional Aesthetics.
		Bar of Counts	Yes	Yes	N/A	Yes with limitation	Only Yes for the categorical variable in Optional Aesthetics.
		Distribution	Bar	N/A	Yes	Yes	
Boxplot	Graphboard	Boxplot	Yes	No	N/A	Yes with limitation	Only Yes on Windows.
		Boxplot	Yes	Yes	N/A	No	
Clustered Boxplot	Graphboard	Clustered Boxplot	Yes	No	N/A	Yes with limitation	Only Yes on Windows.
		Clustered Boxplot	Yes	Yes	N/A	No	
Simple Scatter	Graphboard	Bubble Plot	Yes	No	N/A	Yes with limitation	Only Yes for the continuous variables in both X and Y fields, and the categorical variable in Sizes.
		Scatterplot	Yes	No	N/A	Yes with limitation	Only Yes for the continuous variables in both X and Y fields.
		Plot	Point	N/A	No	Yes with limitation	Only Yes for the continuous variables in both X and Y fields.
Grouped Scatter	Graphboard	Bubble Plot	Yes	Yes	N/A	No	
		Scatterplot	Yes	Yes	N/A	Yes with limitation	Only Yes for the continuous variables in both X and Y fields, and the categorical variable in Optional Aesthetics.
		Plot	Point	N/A	Yes	Yes with limitation	Only Yes for the continuous variables in both X and Y fields, and the categorical variable in the Overlay option.

Exporting graphs

The Export Graph option enables you to export the graph in one of the following formats: Bitmap (.bmp), JPEG (.jpg), PNG (.png), HTML (.html), PDF (.pdf), or ViZml document (.xml) for use in other applications.

Note: When the PDF option is selected, the graphs are exported as high resolution PDF files that are cropped to the size of the graphic. To export graphs, choose File > Export Graph from the menus and then choose the format.

Exporting tables

The Export Table option enables you to export the table in one of the following formats: tab delimited (.tab), comma delimited (.csv), or HTML (.html).

To export tables, choose File > Export Table from the menus and then choose the format.

The remainder of this section focuses on the specific options for creating graphs and using them in their output windows.

- [Setting Header and Footer Preferences](#)
-

Setting Header and Footer Preferences

For graphs and other types of output, you can specify options for headers and footers when the page is printed.

Page Header. Select the type and position of page headers for printing and exporting.

Page Footer. Select the type and position of page footers for printing and exporting.

Font options. Select a font and its point size from the drop-down lists. Click the bold or italic buttons to add these effects.

Draw border around header and footer. Select to enclose the header and footer in a thin rectangular border.

Output Nodes

- [Overview of Output Nodes](#)
 - [Managing output](#)
 - [Viewing output](#)
 - [Table node](#)
 - [Matrix node](#)
 - [Analysis Node](#)
 - [Data Audit node](#)
 - [Transform Node](#)
 - [Statistics Node](#)
 - [Means Node](#)
 - [Report Node](#)
 - [Set Globals Node](#)
 - [Simulation Fitting Node](#)
 - [Simulation Evaluation Node](#)
 - [Extension Output node](#)
 - [KDE nodes](#)
 - [IBM SPSS Statistics Helper Applications](#)
-

Overview of Output Nodes

Output nodes provide the means to obtain information about your data and models. They also provide a mechanism for exporting data in various formats to interface with your other software tools.

The following output nodes are available:

	The Table node displays the data in table format, which can also be written to a file. This is useful anytime that you need to inspect your data values or export them in an easily readable form.
	The Matrix node creates a table that shows relationships between fields. It is most commonly used to show the relationship between two symbolic fields, but it can also show relationships between flag fields or numeric fields.

	The Analysis node evaluates predictive models' ability to generate accurate predictions. Analysis nodes perform various comparisons between predicted values and actual values for one or more model nuggets. They can also compare predictive models to each other.
	The Data Audit node provides a comprehensive first look at the data, including summary statistics, histograms and distribution for each field, as well as information on outliers, missing values, and extremes. Results are displayed in an easy-to-read matrix that can be sorted and used to generate full-size graphs and data preparation nodes.
	The Transform node allows you to select and visually preview the results of transformations before applying them to selected fields.
	The Statistics node provides basic summary information about numeric fields. It calculates summary statistics for individual fields and correlations between fields.
	The Means node compares the means between independent groups or between pairs of related fields to test whether a significant difference exists. For example, you could compare mean revenues before and after running a promotion or compare revenues from customers who did not receive the promotion with those who did.
	The Report node creates formatted reports containing fixed text as well as data and other expressions derived from the data. You specify the format of the report using text templates to define the fixed text and data output constructions. You can provide custom text formatting by using HTML tags in the template and by setting options on the Output tab. You can include data values and other conditional output by using CLEM expressions in the template.
	The Set Globals node scans the data and computes summary values that can be used in CLEM expressions. For example, you can use this node to compute statistics for a field called <i>age</i> and then use the overall mean of <i>age</i> in CLEM expressions by inserting the function <code>@GLOBAL_MEAN(age)</code> .
	The Simulation Fitting node examines the statistical distribution of the data in each field and generates (or updates) a Simulation Generate node, with the best fitting distribution assigned to each field. The Simulation Generate node can then be used to generate simulated data.
	The Simulation Evaluation node evaluates a specified predicted target field, and presents distribution and correlation information about the target field.

Managing output

The Output manager shows the charts, graphs, and tables generated during an IBM® SPSS® Modeler session. You can always reopen an output by double-clicking it in the manager—you do not have to rerun the corresponding stream or node.

To view the Output manager

Open the View menu and choose Managers. Click the Outputs tab.

From the Output manager, you can:

- Display existing output objects, such as histograms, evaluation charts, and tables.
- Rename output objects.
- Save output objects to disk or to the IBM SPSS Collaboration and Deployment Services Repository (if available).
- Add output files to the current project.
- Delete unsaved output objects from the current session.
- Open saved output objects or retrieve them from the IBM SPSS Collaboration and Deployment Services Repository (if available).

To access these options, right-click anywhere on the Outputs tab.

Related information

- [Viewing output](#)

Viewing output

On-screen output is displayed in an output browser window. The output browser window has its own set of menus that allow you to print or save the output, or export it to another format. Note that specific options may vary depending on the type of output.

Printing, saving, and exporting data. More information is available as follows:

- To print the output, use the Print menu option or button. Before you print, you can use Page Setup and Print Preview to set print options and preview the output.
For more information about configuring page headers and footers, see [Setting Header and Footer Preferences](#).
- To save the output to an IBM® SPSS® Modeler output file (.cou), choose Save or Save As from the File menu.
- To save the output in another format, such as text or HTML, choose Export from the File menu. See the topic [Exporting Output](#) for more information.
Note that you can only select these formats if the output contains data that can be sensibly exported in that way. For example, the contents of a decision tree could be exported as text, but the contents of a K-means model would not make sense as text.
- To save the output in a shared repository so that other users can view it using the IBM SPSS Collaboration and Deployment Services Deployment Portal, choose Publish to Web from the File menu. Note that this option requires a separate license for IBM SPSS Collaboration and Deployment Services.

Selecting cells and columns. The Edit menu contains various options for selecting, deselecting, and copying cells and columns, as appropriate for the current output type. See [Selecting cells and columns](#) for more information.

Generating new nodes. The Generate menu enables you to generate new nodes based on the contents of the output browser. The options vary depending on the type of output and the items in the output that are currently selected. For details about the node-generation options for a particular type of output, see the documentation for that output.

- [Publish to Web](#)
- [Viewing Output in an HTML Browser](#)
- [Exporting Output](#)
- [Selecting cells and columns](#)

Publish to Web

The Publish to Web feature enables you to publish certain types of stream output to a central shared IBM® SPSS® Collaboration and Deployment Services Repository that forms the basis of IBM SPSS Collaboration and Deployment Services. If you use this option, other users who need to view this output can do so by using Internet access and an IBM SPSS Collaboration and Deployment Services account--they do not need to have IBM SPSS Modeler installed.

The following table lists the IBM SPSS Modeler nodes that support the Publish to Web feature. Output from these nodes is stored in the IBM SPSS Collaboration and Deployment Services Repository in output object (.cou) format, and can be viewed directly in the IBM SPSS Collaboration and Deployment Services Deployment Portal.

Other types of output can be viewed only if the relevant application (e.g. IBM SPSS Modeler, for stream objects) is installed on the user's machine.

Table 1. Nodes supporting Publish to Web

Node Type	Node
Graphs	all
Output	Table
	Matrix
	Data Audit
	Transform
	Means
	Analysis
	Statistics
	Report (HTML)
IBM SPSS Statistics	Statistics Output

- [Publishing Output to the Web](#)
- [Viewing Published Output Over the Web](#)

Publishing Output to the Web

To publish output to the Web:

1. In an IBM® SPSS® Modeler stream, execute one of the nodes listed in the table. Doing so creates an output object (for example, a table, matrix or report object) in a new window.

2. From the output object window, choose:

File > Publish to Web

Note: If you just want to export simple HTML files for use with a standard Web browser, choose Export from the File menu and select HTML.

3. Connect to the IBM SPSS Collaboration and Deployment Services Repository.

When you have connected successfully, the Repository: Store dialog is displayed, offering a number of storage options.

4. When you have chosen the storage options you want, click Store.

Viewing Published Output Over the Web

You must have an IBM® SPSS® Collaboration and Deployment Services account set up in order to use this feature. If you have the relevant application installed for the object type you want to view (for example, IBM SPSS Modeler or IBM SPSS Statistics), the output is displayed in the application itself rather than in the browser.

To view published output over the Web:

1. Point your browser to `http://<repos_host>:<repos_port>/peb`
where `repos_host` and `repos_port` are the hostname and port number for the IBM SPSS Collaboration and Deployment Services host.
2. Enter the login details for your IBM SPSS Collaboration and Deployment Services account.
3. Click on Content Repository.
4. Navigate to, or search for, the object you want to view.
5. Click on the object name. For some object types, such as graphs, there may be a delay while the object is rendered in the browser.

Viewing Output in an HTML Browser

From the Advanced tab on the Linear, Logistic, and PCA/Factor model nuggets, you can view the displayed information in a separate browser, such as Internet Explorer. The information is output as HTML, enabling you to save it and reuse it elsewhere, such as on a corporate intranet, or Internet site.

To display the information in a browser, click the launch button, situated below the model icon in the top left of the Advanced tab of the model nugget.

Related information

- [Viewing output](#)
- [Exporting Output](#)
- [Selecting cells and columns](#)

Exporting Output

In the output browser window, you may choose to export the output to another format, such as text or HTML. The export formats vary depending on the type of output, but in general are similar to the file type options available if you select Save to file in the node used to generate the output.

Note: You can only select these formats if the output contains data that can be sensibly exported in that way. For example, the contents of a decision tree could be exported as text, but the contents of a K-means model would not make sense as text.

To Export Output

1. In the output browser, open the File menu and choose Export. Then choose the file type that you want to create:

- **Tab Delimited (*.tab).** This option generates a formatted text file containing the data values. This style is often useful for generating a plain-text representation of the information that can be imported into other applications. This option is available for the Table, Matrix, and Means nodes.
- **Comma Delimited (*.dat).** This option generates a comma-delimited text file containing the data values. This style is often useful as a quick way to generate a data file that can be imported into spreadsheets or other data analysis applications. This option is available for the Table, Matrix, and Means nodes.
- **Transposed Tab Delimited (*.tab).** This option is identical to the Tab Delimited option, but the data is transposed so that rows represent fields and the columns represent records.
- **Transposed Comma Delimited (*.dat).** This option is identical to the Comma Delimited option, but the data is transposed so that rows represent fields and the columns represent records.

- **HTML (*.html).** This option writes HTML-formatted output to a file or files.

Related information

- [Viewing output](#)
 - [Viewing Output in an HTML Browser](#)
 - [Selecting cells and columns](#)
-

Selecting cells and columns

A number of nodes, including the Table node, Matrix node, and Means node, generate tabular output. These output tables can be viewed and manipulated in similar ways, including selecting cells, copying all or part of the table to the Clipboard, generating new nodes based on the current selection, and saving and printing the table.

Selecting cells. To select a cell, click it. To select a rectangular range of cells, click one corner of the desired range, drag the mouse to the other corner of the range, and release the mouse button. To select an entire column, click the column heading. To select multiple columns, use Shift-click or Ctrl-click on column headings.

When you make a new selection, the old selection is cleared. By holding down the Ctrl key while selecting, you can add the new selection to any existing selection instead of clearing the old selection. You can use this method to select multiple, noncontiguous regions of the table. The Edit menu also contains the Select All and Clear Selection options.

Reordering columns. The Table node and Means node output browsers allow you to move columns in the table by clicking a column heading and dragging it to the desired location. You can move only one column at a time.

Related information

- [Viewing output](#)
 - [Viewing Output in an HTML Browser](#)
 - [Exporting Output](#)
-

Table node

The Table node creates a table that lists the values in your data. All fields and all values in the stream are included, making this an easy way to inspect your data values or export them in an easily readable form. Optionally, you can highlight records that meet a certain condition.

Note: Unless you are working with small datasets, it is recommended that you select a subset of the data to pass into the Table node. The Table node cannot display properly when the number of records surpasses a size that can be contained in the display structure (for example, 100 million rows).

- [Table Node Settings Tab](#)
 - [Output Node Output Tab](#)
 - [Table Browser](#)
-

Table Node Settings Tab

Highlight records where. You can highlight records in the table by entering a CLEM expression that is true for the records to be highlighted. This option is enabled only when Output to screen is selected.

Related information

- [Table node](#)
 - [Table Browser](#)
 - [Output Node Output Tab](#)
-

Output Node Output Tab

For nodes that generate table-style output, the Output tab enables you to specify the format and location of the results.

Output name. Specifies the name of the output produced when the node is executed. Auto chooses a name based on the node that generates the output. Optionally, you can select Custom to specify a different name.

Output to screen (the default). Creates an output object to view online. The output object will appear on the Outputs tab of the manager window when the output node is executed.

Output to file. Saves the output to a file when the node is executed. If you choose this option, enter a filename (or navigate to a directory and specify a filename using the File Chooser button) and select a file type. Note that some file types may be unavailable for certain types of output.

Note:

Output data from output nodes is encoded according to the following rules:

- When executing an output node, the stream encoding value (which is set on the Stream Options tab) will be set to the output.
- After the output is generated, its encoding will not be changed even when the stream encoding is changed.
- When exporting output node output, the output file is exported with the currently defined stream encoding. After the output is created, even if you change the stream encoding, it will not impact the generated output.

Note the following exceptions to these rules:

- All HTML exports are encoded in UTF-8 format.
- Output from the Extension output node is generated by a custom user script. So the encoding is controlled by the script.

The following options are available for saving the output to a file:

- Data (tab delimited) (*.tab). This option generates a formatted text file containing the data values. This style is often useful for generating a plain-text representation of the information that can be imported into other applications. This option is available for the Table, Matrix, and Means nodes.
- Data (comma delimited) (*.dat). This option generates a comma-delimited text file containing the data values. This style is often useful as a quick way to generate a data file that can be imported into spreadsheets or other data analysis applications. This option is available for the Table, Matrix, and Means nodes.
- HTML (*.html). This option writes HTML-formatted output to a file or files. For tabular output (from the Table, Matrix, or Means nodes), a set of HTML files contains a contents panel listing field names and the data in an HTML table. The table may be split over multiple HTML files if the number of rows in the table exceeds the Lines per page specification. In this case, the contents panel contains links to all table pages and provides a means of navigating the table. For non-tabular output, a single HTML file is created containing the results of the node.
Note: If the HTML output contains only formatting for the first page, select Paginate output and adjust the Lines per page specification to include all output on a single page. Or if the output template for nodes such as the Report node contains custom HTML tags, be sure you have specified Custom as the format type.
- Text File (*.txt). This option generates a text file containing the output. This style is often useful for generating output that can be imported into other applications, such as word processors or presentation software. This option is not available for some nodes.
- Output object (*.cou). Output objects saved in this format can be opened and viewed in IBM® SPSS® Modeler, added to projects, and published and tracked using the IBM SPSS Collaboration and Deployment Services Repository.

Output view. For the Means node, you can specify whether simple or advanced output is displayed by default. Note you can also toggle between these views when browsing the generated output. See the topic [Means Node Output Browser](#) for more information.

Format. For the Report node, you can choose whether output is automatically formatted or formatted using HTML included in the template. Select Custom to allow HTML formatting in the template.

Title. For the Report node, you can specify optional title text that will appear at the top of the report output.

Highlight inserted text. For the Report node, select this option to highlight text generated by CLEM expressions in the Report template. See the topic [Report Node Template Tab](#) for more information. This option is not recommended when using Custom formatting.

Lines per page. For the Report node, specify a number of lines to include on each page during Auto formatting of the output report.

Transpose data. This option transposes the data before export, so that rows represent fields and the columns represent records.

Note: For large tables, the above options can be somewhat inefficient, especially when working with a remote server. In such cases, using a File output node provides much better performance. See the topic [Flat File Export Node](#) for more information.

Table Browser

The table browser displays tabular data and enables you to perform standard operations including selecting and copying cells, reordering columns, and saving and printing the table. See the topic [Selecting cells and columns](#) for more information. These are the same operations that you can carry out when previewing the data in a node.

Exporting table data. You can export data from the table browser by choosing:

File...>Export

See the topic [Exporting Output](#) for more information.

Data is exported in the system default encoding format, which is specified in the Windows Control Panel or, if running in distributed mode, on the server computer.

Searching the table. The search button (with the binoculars icon) on the main toolbar activates the search toolbar, enabling you to search the table for specific values. You can search forward or backward in the table, you can specify a case-sensitive search (the Aa button), and you can interrupt a search-in-progress with the interrupt search button.

Generating new nodes. The Generate menu contains node generation operations.

- **Select Node ("Records").** Generates a Select node that selects the records for which any cell in the table is selected.
- **Select ("And").** Generates a Select node that selects records containing *all* of the values selected in the table.
- **Select ("Or").** Generates a Select node that selects records containing *any* of the values selected in the table.
- **Derive ("Records").** Generates a Derive node to create a new flag field. The flag field contains *T* for records for which any cell in the table is selected and *F* for the remaining records.
- **Derive ("And").** Generates a Derive node to create a new flag field. The flag field contains *T* for records containing *all* of the values selected in the table and *F* for the remaining records.
- **Derive ("Or").** Generates a Derive node to create a new flag field. The flag field contains *T* for records containing *any* of the values selected in the table and *F* for the remaining records.

Related information

- [Viewing output](#)
 - [Table node](#)
 - [Table Node Settings Tab](#)
-

Matrix node

The Matrix node enables you to create a table that shows relationships between fields. It is most commonly used to show the relationship between two categorical fields (flag, nominal, or ordinal), but it can also be used to show relationships between continuous (numeric range) fields.

- [Matrix Node Settings Tab](#)
 - [Matrix Node Appearance Tab](#)
 - [Matrix node output browser](#)
-

Matrix Node Settings Tab

The Settings tab enables you to specify options for the structure of the matrix.

Fields. Select a field selection type from the following options:

- **Selected.** This option enables you to select a categorical field for the rows and one for the columns of the matrix. The rows and columns of the matrix are defined by the list of values for the selected categorical field. The cells of the matrix contain the summary statistics selected below.
- **All flags (true values).** This option requests a matrix with one row and one column for each flag field in the data. The cells of the matrix contain the counts of double positives for each flag combination. In other words, for a row corresponding to *bought bread* and a column corresponding to *bought cheese*, the cell at the intersection of that row and column contains the number of records for which both *bought bread* and *bought cheese* are true.
- **All numerics.** This option requests a matrix with one row and one column for each numeric field. The cells of the matrix represent the sum of the cross-products for the corresponding pair of fields. In other words, for each cell in the matrix, the values for the row field and the column field are multiplied for each record and then summed across records.

Include missing values. Includes user-missing (blank) and system missing (\$null\$) values in the row and column output. For example, if the value *N/A* has been defined as user-missing for the selected column field, a separate column labeled *N/A* will be included in the table (assuming this value actually occurs in the data) just like any other category. If this option is deselected, the *N/A* column is excluded regardless of how often it occurs.

Note: The option to include missing values applies only when selected fields are cross-tabulated. Blank values are mapped to \$null\$ and are excluded from aggregation for the function field when the mode is Selected and the content is set to Function and for all numeric fields when the mode is set to All Numerics.

Cell contents. If you have chosen Selected fields above, you can specify the statistic to be used in the cells of the matrix. Select a count-based statistic, or select an overlay field to summarize values of a numeric field based on the values of the row and column fields.

- **Cross-tabulations.** Cell values are counts and/or percentages of how many records have the corresponding combination of values. You can specify which cross-tabulation summaries you want using the options on the Appearance tab. The global chi-square value is also displayed along with the significance. See the topic [Matrix node output browser](#) for more information.
- **Function.** If you select a summary function, cell values are a function of the selected overlay field values for cases having the appropriate row and column values. For example, if the row field is *Region*, the column field is *Product*, and the overlay field is *Revenue*, then the cell in the *Northeast* row and the *Widget* column will contain the sum (or average, minimum, or maximum) of revenue for widgets sold in the northeast region. The default summary function is Mean. You can select another function for summarizing the function field. Options include Mean, Sum, SDev (standard deviation), Max (maximum), and Min (minimum).

Related information

- [Matrix node](#)
 - [Matrix Node Appearance Tab](#)
 - [Matrix node output browser](#)
-

Matrix Node Appearance Tab

The Appearance tab enables you to control sorting and highlighting options for the matrix, as well as statistics presented for cross-tabulation matrices.

Rows and columns. Controls the sorting of row and column headings in the matrix. The default is Unsorted. Select Ascending or Descending to sort row and column headings in the specified direction.

Overlay. Enables you to highlight extreme values in the matrix. Values are highlighted based on cell counts (for cross-tabulation matrices) or calculated values (for function matrices).

- **Highlight top.** You can request the highest values in the matrix to be highlighted (in red). Specify the number of values to highlight.
- **Highlight bottom.** You can also request the lowest values in the matrix to be highlighted (in green). Specify the number of values to highlight.

Note: For the two highlighting options, ties can cause more values than requested to be highlighted. For example, if you have a matrix with six zeros among the cells and you request Highlight bottom 5, all six zeros will be highlighted.

Cross-tabulation cell contents. For cross-tabulations, you can specify the summary statistics contained in the matrix for cross-tabulation matrices. These options are not available when either the All Numerics or Function option is selected on the Settings tab.

- **Counts.** Cells include the number of records with the row value that have the corresponding column value. This is the only default cell content.
- **Expected values.** The expected value for number of records in the cell, assuming that there is no relationship between the rows and columns. Expected values are based on the following formula:
$$p(\text{row value}) * p(\text{column value}) * \text{total number of records}$$
- **Residuals.** The difference between observed and expected values.
- **Percentage of row.** The percentage of all records with the row value that have the corresponding column value. Percentages sum to 100 within rows.
- **Percentage of column.** The percentage of all records with the column value that have the corresponding row value. Percentages sum to 100 within columns.
- **Percentage of total.** The percentage of all records having the combination of column value and row value. Percentages sum to 100 over the whole matrix.
- **Include row and column totals.** Adds a row and a column to the matrix for column and row totals.
- **Apply Settings.** (Output Browser only) Enables you to make changes to the appearance of the Matrix node output without having to close and reopen the Output Browser. Make the changes on this tab of the Output Browser, click this button and then select the Matrix tab to see the effect of the changes.

Related information

- [Matrix node](#)
 - [Matrix Node Settings Tab](#)
 - [Matrix node output browser](#)
 - [Output Node Output Tab](#)
-

Matrix node output browser

The matrix browser displays cross-tabulated data and enables you to perform operations on the matrix, including selecting cells, copying the matrix to the Clipboard in whole or in part, generating new nodes based on the matrix selection, and saving and printing the matrix. The matrix browser may also be used to display output from certain models, such as Naive Bayes models from Oracle.

The File and Edit menus provide the usual options for printing, saving, exporting output, and for selecting and copying data. See the topic [Viewing output](#) for more information.

Chi-square. For a cross-tabulation of two categorical fields, the global Pearson chi-square is also displayed below the table. This test indicates the probability that the two fields are unrelated, based on the difference between observed counts and the counts you would expect if no relationship exists. For example, if there were no relationship between customer satisfaction and store location, you would expect similar satisfaction rates across all stores. But if customers at certain stores consistently report higher rates than others, you might suspect it wasn't a coincidence. The greater the difference, the smaller the probability that it was the result of chance sampling error alone.

- The chi-square test indicates the probability that the two fields are unrelated, in which case any differences between observed and expected frequencies are the result of chance alone. If this probability is very small—typically less than 5%—then the relationship between the two fields is said to be significant.
- If there is only one column or one row (a one-way chi-square test), the degrees of freedom is the number of cells minus one. For a two-way chi-square, the degrees of freedom is the number of rows minus one times the number of columns minus one.
- Use caution when interpreting the chi-square statistic if any of the expected cell frequencies are less than five.
- The chi-square test is available only for a cross-tabulation of two fields. (When All flags or All numerics is selected on the Settings tab, this test is not displayed.)

Generate menu. The Generate menu contains node generation operations. These operations are available only for cross-tabulated matrices, and you must have at least one cell selected in the matrix.

- Select Node. Generates a Select node that selects the records that match any selected cell in the matrix.
- Derive Node (Flag). Generates a Derive node to create a new flag field. The flag field contains *T* for records that match any selected cell in the matrix and *F* for the remaining records.
- Derive Node (Set). Generates a Derive node to create a new nominal field. The nominal field contains one category for each contiguous set of selected cells in the matrix.

Related information

- [Viewing output](#)
- [Matrix node](#)
- [Matrix Node Settings Tab](#)
- [Matrix Node Appearance Tab](#)

Analysis Node

The Analysis node enables you to evaluate the ability of a model to generate accurate predictions. Analysis nodes perform various comparisons between predicted values and actual values (your target field) for one or more model nuggets. Analysis nodes can also be used to compare predictive models to other predictive models.

When you execute an Analysis node, a summary of the analysis results is automatically added to the Analysis section on the Summary tab for each model nugget in the executed stream. The detailed analysis results appear on the Outputs tab of the manager window or can be written directly to a file.

Note: Because Analysis nodes compare predicted values to actual values, they are only useful with supervised models (those that require a target field). For unsupervised models such as clustering algorithms, there are no actual results available to use as a basis for comparison.

- [Analysis Node Analysis Tab](#)
- [Analysis Output Browser](#)

Analysis Node Analysis Tab

The Analysis tab enables you to specify the details of the analysis.

Coincidence matrices (for symbolic or categorical targets). Shows the pattern of matches between each generated (predicted) field and its target field for categorical targets (either flag, nominal, or ordinal). A table is displayed with rows defined by actual values and columns defined by predicted values, with the number of records having that pattern in each cell. This is useful for identifying systematic errors in prediction. If there is more than one generated field related to the same output field but produced by different models, the cases where these fields agree and disagree are counted and the totals are displayed. For the cases where they agree, another set of correct/wrong statistics is displayed.

Performance evaluation. Shows performance evaluation statistics for models with categorical outputs. This statistic, reported for each category of the output field(s), is a measure of the average information content (in bits) of the model for predicting records belonging to that category. It takes the difficulty of the classification problem into account, so accurate predictions for rare categories will earn a higher performance evaluation index than accurate predictions for common categories. If the model does no better than guessing for a category, the performance evaluation index for that category will be 0.

Evaluation metrics (AUC & Gini, binary classifiers only). For binary classifiers, this option reports the AUC (area under curve) and Gini coefficient evaluation metrics. Both of these evaluation metrics are calculated together for each binary model. The values of the metrics are reported in a table in the analysis output browser.

The AUC evaluation metric is calculated as the area under an ROC (receiver operator characteristic) curve, and is a scalar representation of the expected performance of a classifier. The AUC is always between 0 and 1, with a higher number representing a better classifier. A diagonal ROC curve between the coordinates (0,0) and (1,1) represents a random classifier, and has an AUC of 0.5. Therefore, a realistic classifier will not have an AUC of less than 0.5.

The Gini coefficient evaluation metric is sometimes used as an alternative evaluation metric to the AUC, and the two measures are closely related. The Gini coefficient is calculated as twice the area between the ROC curve and the diagonal, or as $\text{Gini} = 2\text{AUC} - 1$. The Gini coefficient is always between 0 and 1, with a higher number representing a better classifier. The Gini coefficient is negative in the unlikely event that the ROC curve is below the diagonal.

Confidence figures (if available). For models that generate a confidence field, this option reports statistics on the confidence values and their relationship to predictions. There are two settings for this option:

- **Threshold for.** Reports the confidence level above which the accuracy will be the specified percentage.
- **Improve accuracy.** Reports the confidence level above which the accuracy is improved by the specified factor. For example, if the overall accuracy is 90% and this option is set to 2.0, the reported value will be the confidence required for 95% accuracy.

Find predicted/predictor fields using. Determines how predicted fields are matched to the original target field.

- **Model output field metadata.** Matches predicted fields to the target based on model field information, allowing a match even if a predicted field has been renamed. Model field information can also be accessed for any predicted field from the Values dialog box using a Type node. See the topic [Using the Values Dialog Box](#) for more information.
- **Field name format.** Matches fields based on the naming convention. For example predicted values generated by a C5.0 model nugget for a target named *response* must be in a field named *\$C-response*.

Separate by partition. If a partition field is used to split records into training, test, and validation samples, select this option to display results separately for each partition. See the topic [Partition Node](#) for more information.

Note: When separating by partition, records with null values in the partition field are excluded from the analysis. This will never be an issue if a Partition node is used, since Partition nodes do not generate null values.

User defined analysis. You can specify your own analysis calculation to be used in evaluating your model(s). Use CLEM expressions to specify what should be computed for each record and how to combine the record-level scores into an overall score. Use the functions @TARGET and @PREDICTED to refer to the target (actual output) value and the predicted value, respectively.

- **If.** Specify a conditional expression if you need to use different calculations depending on some condition.
- **Then.** Specify the calculation if the If condition is true.
- **Else.** Specify the calculation if the If condition is false.
- **Use.** Select a statistic to compute an overall score from the individual scores.

Break down analysis by fields. Shows the categorical fields available for breaking down the analysis. In addition to the overall analysis, a separate analysis will be reported for each category of each breakdown field.

Related information

- [Analysis Node](#)
- [Analysis Output Browser](#)
- [Output Node Output Tab](#)

Analysis Output Browser

The analysis output browser displays the results of executing the Analysis node. The usual saving, exporting, and printing options are available from the File menu. See the topic [Viewing output](#) for more information.

When you first browse Analysis output, the results are expanded. To hide results after viewing them, use the expander control to the left of the item to collapse the specific results you want to hide or click the Collapse All button to collapse all results. To see results again after collapsing them, use the expander control to the left of the item to show the results or click the Expand All button to show all results.

Results for output field. The Analysis output contains a section for each output field for which there is a corresponding prediction field created by a generated model.

Comparing. Within the output field section is a subsection for each prediction field associated with that output field. For categorical output fields, the top level of this section contains a table showing the number and percentage of correct and incorrect predictions and the total number of records in the stream. For numeric output fields, this section shows the following information:

- **Minimum Error.** Shows the minimum error (difference between observed and predicted values).
- **Maximum Error.** Shows the maximum error.
- **Mean Error.** Shows the average (mean) of errors across all records. This indicates whether there is a systematic **bias** (a stronger tendency to overestimate than to underestimate, or vice versa) in the model.
- **Mean Absolute Error.** Shows the average of the absolute values of the errors across all records. Indicates the average magnitude of error, independent of the direction.
- **Standard Deviation.** Shows the standard deviation of the errors.
- **Linear Correlation.** Shows the linear correlation between the predicted and actual values. This statistic varies between -1.0 and 1.0. Values close to +1.0 indicate a strong positive association, so that high predicted values are associated with high actual values and low predicted values are associated with low actual values. Values close to -1.0 indicate a strong negative association, so that high predicted values are associated with low actual values, and vice versa. Values close to 0.0 indicate a weak association, so that predicted values are more or less independent of actual values. Note: A blank entry here indicates that linear correlation cannot be computed in this case, because either the actual or predicted values are constant.
- **Occurrences.** Shows the number of records used in the analysis.

Coincidence Matrix. For categorical output fields, if you requested a coincidence matrix in the analysis options, a subsection appears here containing the matrix. The rows represent actual observed values, and the columns represent predicted values. The cell in the table indicates the number of records for each combination of predicted and actual values.

Performance Evaluation. For categorical output fields, if you requested performance evaluation statistics in the analysis options, the performance evaluation results appear here. Each output category is listed with its performance evaluation statistic.

Confidence Values Report. For categorical output fields, if you requested confidence values in the analysis options, the values appear here. The following statistics are reported for model confidence values:

- **Range.** Shows the range (smallest and largest values) of confidence values for records in the stream data.
- **Mean Correct.** Shows the average confidence for records that are classified correctly.
- **Mean Incorrect.** Shows the average confidence for records that are classified incorrectly.
- **Always Correct Above.** Shows the confidence threshold above which predictions are always correct and shows the percentage of cases meeting this criterion.
- **Always Incorrect Below.** Shows the confidence threshold below which predictions are always incorrect and shows the percentage of cases meeting this criterion.
- **X% Accuracy Above.** Shows the confidence level at which accuracy is X%. X is approximately the value specified for Threshold for in the Analysis options. For some models and datasets, it is not possible to choose a confidence value that gives the exact threshold specified in the options (usually due to clusters of similar cases with the same confidence value near the threshold). The threshold reported is the closest value to the specified accuracy criterion that can be obtained with a single confidence value threshold.
- **X Fold Correct Above.** Shows the confidence value at which accuracy is X times better than it is for the overall dataset. X is the value specified for Improve accuracy in the Analysis options.

Agreement between. If two or more generated models that predict the same output field are included in the stream, you will also see statistics on the **agreement** between predictions generated by the models. This includes the number and percentage of records for which the predictions agree (for categorical output fields) or error summary statistics (for continuous output fields). For categorical fields, it includes an analysis of predictions compared to actual values for the subset of records on which the models agree (generate the same predicted value).

Evaluation Metrics. For binary classifiers, if you requested evaluation metrics in the analysis options, the values of the AUC and Gini coefficient evaluation metrics are shown in a table in this section. The table has one row for each binary classifier model. The evaluation metrics table is shown for each output field, rather than for each model.

Related information

- [Viewing output](#)
 - [Analysis Node](#)
 - [Analysis Node Analysis Tab](#)
-

Data Audit node

The Data Audit node provides a comprehensive first look at the data you bring into IBM® SPSS® Modeler, presented in an easy-to-read matrix that can be sorted and used to generate full-size graphs and a variety of data preparation nodes.

- The Audit tab displays a report that provides summary statistics, histograms, and distribution graphs that may be useful in gaining a preliminary understanding of the data. The report also displays the storage icon before the field name.
- The Quality tab in the audit report displays information about outliers, extremes, and missing values, and offers tools for handling these values.

Using the Data Audit node

The Data Audit node can be attached directly to a source node or downstream from an instantiated Type node. You can also generate a number of data preparation nodes based on the results. For example, you can generate a Filter node that excludes fields with too many missing values to be useful in modeling, and generate a SuperNode that imputes missing values for any or all of the fields that remain. This is where the real power of the audit comes in, enabling you not only to assess the current state of your data, but to take action based on the assessment.

Screening or sampling the data. Because an initial audit is particularly effective when dealing with big data, a Sample node may be used to reduce processing time during the initial exploration by selecting only a subset of records. The Data Audit node can also be used in combination with nodes such as Feature Selection and Anomaly Detection in the exploratory stages of analysis.

- [Data Audit Node Settings Tab](#)
- [Data Audit Quality Tab](#)
- [Data Audit Output Browser](#)

Data Audit Node Settings Tab

The Settings tab enables you to specify basic parameters for the audit.

Default. You can simply attach the node to your stream and click Run to generate an audit report for all fields based on default settings, as follows:

- If there are no Type node settings, all fields are included in the report.
- If there are Type settings (regardless of whether or not they are instantiated), all *Input*, *Target*, and *Both* fields are included in the display. If there is a single *Target* field, use it as the Overlay field. If there is more than one *Target* field specified, no default overlay is specified.

Use custom fields. Select this option to manually select fields. Use the field chooser button on the right to select fields individually or by type.

Overlay field. The overlay field is used in drawing the thumbnail graphs shown in the audit report. In the case of a continuous (numeric range) field, bivariate statistics (covariance and correlation) are also calculated. If a single *Target* field is present based on Type node settings, it is used as the default overlay field as described above. Alternatively, you can select Use custom fields in order to specify an overlay.

Display. Enables you to specify whether graphs are available in the output, and to choose the statistics displayed by default.

- Graphs. Displays a graph for each selected field; either a distribution (bar) graph, histogram, or scatterplot as appropriate for the data. Graphs are displayed as thumbnails in the initial report, but full-sized graphs and graph nodes can also be generated. See the topic [Data Audit Output Browser](#) for more information.
- Basic/Advanced statistics. Specifies the level of statistics displayed in the output by default. While this setting determines the initial display, all statistics are available in the output regardless of this setting. See the topic [Display Statistics](#) for more information.

Median and mode. Calculates the median and mode for all fields in the report. Note that with large datasets, these statistics may increase processing time, since they take longer than others to compute. In the case of the median only, the reported value may be based on a sample of 2000 records (rather than the full dataset) in some cases. This sampling is done on a per-field basis in cases where memory limits would otherwise be exceeded. When sampling is in effect, the results will be labeled as such in the output (*Sample Median* rather than just *Median*). All statistics other than the median are always computed using the full dataset.

Empty or typeless fields. When used with instantiated data, typeless fields are not included in the audit report. To include typeless fields (including empty fields), select Clear All Values in any upstream Type nodes. This ensures that data are not instantiated, causing all fields to be included in the report. For example, this may be useful if you want to obtain a complete list of all fields or generate a Filter node that will exclude those that are empty. See the topic [Filtering Fields with Missing Data](#) for more information.

Related information

- [Data Audit node](#)
- [Data Audit Quality Tab](#)
- [Data Audit Output Browser](#)
- [Viewing and Generating Graphs](#)
- [Display Statistics](#)
- [Data Audit Browser Quality Tab](#)
- [Imputing Missing Values](#)
- [Handling Outliers and Extreme Values](#)
- [Filtering Fields with Missing Data](#)

- [Selecting Records with Missing Data](#)
 - [Generating Other Nodes for Data Preparation](#)
-

Data Audit Quality Tab

The Quality tab in the Data Audit node provides options for handling missing values, outliers, and extreme values.

Missing Values

- Count of records with valid values. Select this option to show the number of records with valid values for each evaluated field. Note that null (undefined) values, blank values, white spaces and empty strings are always treated as invalid values.
- Breakdown counts of records with invalid values. Select this option to show the number of records with each type of invalid value for each field.

Outliers and Extreme Values

Detection method for outliers and extreme values. Two methods are supported:

Standard deviation from the mean. Detects outliers and extremes based on the number of standard deviations from the mean. For example, if you have a field with a mean of 100 and a standard deviation of 10, you could specify 3.0 to indicate that any value below 70 or above 130 should be treated as an outlier.

Interquartile range. Detects outliers and extremes based on the interquartile range, which is the range within which the two central quartiles fall (between the 25th and 75th percentiles). For example, based on the default setting of 1.5, the lower threshold for outliers would be $Q1 - 1.5 * IQR$ and the upper threshold would be $Q3 + 1.5 * IQR$. Note that using this option may slow performance on large datasets.

Related information

- [Data Audit node](#)
 - [Data Audit Node Settings Tab](#)
 - [Data Audit Output Browser](#)
 - [Viewing and Generating Graphs](#)
 - [Display Statistics](#)
 - [Data Audit Browser Quality Tab](#)
 - [Imputing Missing Values](#)
 - [Handling Outliers and Extreme Values](#)
 - [Filtering Fields with Missing Data](#)
 - [Selecting Records with Missing Data](#)
 - [Generating Other Nodes for Data Preparation](#)
-

Data Audit Output Browser

The Data Audit browser is a powerful tool for gaining overview of your data. The Audit tab displays thumbnail graphs, storage icons, and statistics for all fields, while the Quality tab displays information about outliers, extremes, and missing values. Based on the initial graphs and summary statistics, you might decide to recode a numeric field, derive a new field, or reclassify the values of a nominal field. Or you may want to explore further using more sophisticated visualization. You can do this right from the audit report browser using the Generate menu to create any number of nodes that can be used to transform or visualize the data.

- Sort columns by clicking on the column header, or reorder columns using drag and drop. Most standard output operations are also supported. See the topic [Viewing output](#) for more information.
- View values and ranges for fields by double-clicking a field in the Measurement or Unique columns.
- Use the toolbar or Edit menu to show or hide value labels, or to choose the statistics you want to display. See the topic [Display Statistics](#) for more information.
- Verify the storage icons to the left of the field names. Storage describes the way data are stored in a field. For example, a field with values of 1 and 0 stores integer data. This is distinct from the measurement level, which describes the usage of the data, and does not affect storage. See the topic [Setting Field Storage and Formatting](#) for more information.
- [Viewing and Generating Graphs](#)
- [Display Statistics](#)
- [Data Audit Browser Quality Tab](#)
- [Generating Other Nodes for Data Preparation](#)

Related information

- [Data Audit node](#)
 - [Data Audit Node Settings Tab](#)
 - [Data Audit Quality Tab](#)
 - [Viewing and Generating Graphs](#)
 - [Display Statistics](#)
 - [Data Audit Browser Quality Tab](#)
 - [Imputing Missing Values](#)
 - [Handling Outliers and Extreme Values](#)
 - [Filtering Fields with Missing Data](#)
 - [Selecting Records with Missing Data](#)
 - [Generating Other Nodes for Data Preparation](#)
-

Viewing and Generating Graphs

If no overlay is selected, the Audit tab displays either bar charts (for nominal or flag fields) or histograms (continuous fields).

For a nominal or flag field overlay, the graphs are colored by the values of the overlay.

For a continuous field overlay, two-dimensional scatterplots are generated rather than one-dimensional bars and histograms. In this case, the x axis maps to the overlay field, enabling you to see the same scale on all x axes as you read down the table.

- For Flag or Nominal fields, hold the mouse cursor over a bar to display the underlying value or label in a ToolTip.
 - For Flag or Nominal fields, use the toolbar to toggle the orientation of thumbnail graphs from horizontal to vertical.
 - To generate a full-sized graph from any thumbnail, double-click on the thumbnail, or select a thumbnail and choose Graph Output from the Generate menu. *Note:* If a thumbnail graph was based on sampled data, the generated graph will contain all cases if the original data stream is still open.
- You can only generate a graph if the Data Audit node that created the output is connected to the stream.
- To generate a matching graph node, select one or more fields on the Audit tab and choose Graph Node from the Generate menu. The resulting node is added to the stream canvas and can be used to re-create the graph each time the stream is run.
 - If an overlay set has more than 100 values, a warning is raised and the overlay is not included.

Related information

- [Data Audit node](#)
 - [Data Audit Node Settings Tab](#)
 - [Data Audit Quality Tab](#)
 - [Data Audit Output Browser](#)
 - [Display Statistics](#)
 - [Data Audit Browser Quality Tab](#)
 - [Imputing Missing Values](#)
 - [Handling Outliers and Extreme Values](#)
 - [Filtering Fields with Missing Data](#)
 - [Selecting Records with Missing Data](#)
 - [Generating Other Nodes for Data Preparation](#)
-

Display Statistics

The Display Statistics dialog box enables you to choose the statistics displayed on the Audit tab. The initial settings are specified in the Data Audit node. See the topic [Data Audit Node Settings Tab](#) for more information.

Minimum. The smallest value of a numeric variable.

Maximum. The largest value of a numeric variable.

Sum. The sum or total of the values, across all cases with nonmissing values.

Range. The difference between the largest and smallest values of a numeric variable, the maximum minus the minimum.

Mean. A measure of central tendency. The arithmetic average, the sum divided by the number of cases.

Standard Error of Mean. A measure of how much the value of the mean may vary from sample to sample taken from the same distribution. It can be used to roughly compare the observed mean to a hypothesized value (that is, you can conclude the two values are different if the ratio of the difference to the standard error is less than -2 or greater than +2).

standard deviation. A measure of dispersion around the mean, equal to the square root of the variance. The standard deviation is measured in the same units as the original variable.

Variance. A measure of dispersion around the mean, equal to the sum of squared deviations from the mean divided by one less than the number of cases. The variance is measured in units that are the square of those of the variable itself.

Skewness. A measure of the asymmetry of a distribution. The normal distribution is symmetric and has a skewness value of 0. A distribution with a significant positive skewness has a long right tail. A distribution with a significant negative skewness has a long left tail. As a guideline, a skewness value more than twice its standard error is taken to indicate a departure from symmetry.

Standard Error of Skewness. The ratio of skewness to its standard error can be used as a test of normality (that is, you can reject normality if the ratio is less than -2 or greater than +2). A large positive value for skewness indicates a long right tail; an extreme negative value indicates a long left tail.

Kurtosis. A measure of the extent to which there are outliers. For a normal distribution, the value of the kurtosis statistic is zero. Positive kurtosis indicates that the data exhibit more extreme outliers than a normal distribution. Negative kurtosis indicates that the data exhibit less extreme outliers than a normal distribution.

Standard Error of Kurtosis. The ratio of kurtosis to its standard error can be used as a test of normality (that is, you can reject normality if the ratio is less than -2 or greater than +2). A large positive value for kurtosis indicates that the tails of the distribution are longer than those of a normal distribution; a negative value for kurtosis indicates shorter tails (becoming like those of a box-shaped uniform distribution).

Unique. Evaluates all effects simultaneously, adjusting each effect for all other effects of any type.

Valid. Valid cases having neither the system-missing value, nor a value defined as user-missing. Note that null (undefined) values, blank values, white spaces and empty strings are always treated as invalid values.

Median. The value above and below which half of the cases fall, the 50th percentile. If there is an even number of cases, the median is the average of the two middle cases when they are sorted in ascending or descending order. The median is a measure of central tendency not sensitive to outlying values (unlike the mean, which can be affected by a few extremely high or low values).

Mode. The most frequently occurring value. If several values share the greatest frequency of occurrence, each of them is a mode.

Note that median and mode are suppressed by default in order to improve performance but can be selected on the Settings tab in the Data Audit node. See the topic [Data Audit Node Settings Tab](#) for more information.

Statistics for Overlays

If a continuous (numeric range) overlay field is in use, the following statistics are also available:

Covariance. An unstandardized measure of association between two variables, equal to the cross-product deviation divided by N-1.

Related information

- [Data Audit node](#)
 - [Data Audit Node Settings Tab](#)
 - [Data Audit Quality Tab](#)
 - [Data Audit Output Browser](#)
 - [Viewing and Generating Graphs](#)
 - [Data Audit Browser Quality Tab](#)
 - [Imputing Missing Values](#)
 - [Handling Outliers and Extreme Values](#)
 - [Filtering Fields with Missing Data](#)
 - [Selecting Records with Missing Data](#)
 - [Generating Other Nodes for Data Preparation](#)
-

Data Audit Browser Quality Tab

The Quality tab in the Data Audit browser displays the results of the data quality analysis and enables you to specify treatments for outliers, extremes, and missing values.

- [Imputing Missing Values](#)
- [Handling Outliers and Extreme Values](#)
- [Filtering Fields with Missing Data](#)
- [Selecting Records with Missing Data](#)

Related information

- [Data Audit node](#)
 - [Data Audit Node Settings Tab](#)
 - [Data Audit Quality Tab](#)
 - [Data Audit Output Browser](#)
 - [Viewing and Generating Graphs](#)
 - [Display Statistics](#)
 - [Imputing Missing Values](#)
 - [Handling Outliers and Extreme Values](#)
 - [Filtering Fields with Missing Data](#)
 - [Selecting Records with Missing Data](#)
 - [Generating Other Nodes for Data Preparation](#)
-

Imputing Missing Values

The audit report lists the percentage of complete records for each field, along with the number of valid, null, and blank values. You can choose to impute missing values for specific fields as appropriate, and then generate a SuperNode to apply these transformations.

1. In the Impute Missing column, specify the type of values you want to impute, if any. You can choose to impute blanks, nulls, both, or specify a custom condition or expression that selects the values to impute.

There are several types of missing values recognized by IBM® SPSS® Modeler:

- Null or system-missing values. These are nonstring values that have been left blank in the database or source file and have not been specifically defined as "missing" in a source or Type node. System-missing values are displayed as \$null\$. Note that empty strings are not considered nulls in IBM SPSS Modeler, although they may be treated as nulls by certain databases.
- Empty strings and white space. Empty string values and white space (strings with no visible characters) are treated as distinct from null values. Empty strings are treated as equivalent to white space for most purposes. For example, if you select the option to treat white space as blanks in a source or Type node, this setting applies to empty strings as well.
- Blank or user-defined missing values. These are values such as `unknown`, `99`, or `-1` that are explicitly defined in a source node or Type node as missing. Optionally, you can also choose to treat nulls and white space as blanks, which allows them to be flagged for special treatment and to be excluded from most calculations. For example, you can use the `@BLANK` function to treat these values, along with other types of missing values, as blanks.

2. In the Method column, specify the method you want to use.

The following methods are available for imputing missing values:

Fixed. Substitutes a fixed value (either the field mean, midpoint of the range, or a constant that you specify).

Random. Substitutes a random value based on a normal or uniform distribution.

Expression. Allows you to specify a custom expression. For example, you could replace values with a global variable created by the Set Globals node.

Algorithm. Substitutes a value predicted by a model based on the C&RT algorithm. For each field imputed using this method, there will be a separate C&RT model, along with a Filler node that replaces blanks and nulls with the value predicted by the model. A Filter node is then used to remove the prediction fields generated by the model.

3. To generate a Missing Values SuperNode, from the menus choose:

Generate > Missing Values SuperNode

The Missing Values SuperNode dialog box is displayed.

4. Select All fields or Selected fields only, and specify a sample size if desired. (The specified sample is a percentage; by default, 10% of all records are sampled.)
5. Click OK to add the generated SuperNode to the stream canvas.
6. Attach the SuperNode to the stream to apply the transformations.

Within the SuperNode, a combination of model nugget, Filler, and Filter nodes is used as appropriate. To understand how it works, you can edit the SuperNode and click Zoom In, and you can add, edit, or remove specific nodes within the SuperNode to fine-tune the behavior.

Related information

- [Data Audit node](#)
- [Data Audit Node Settings Tab](#)
- [Data Audit Quality Tab](#)
- [Data Audit Output Browser](#)
- [Viewing and Generating Graphs](#)

- [Display Statistics](#)
 - [Data Audit Browser Quality Tab](#)
 - [Handling Outliers and Extreme Values](#)
 - [Filtering Fields with Missing Data](#)
 - [Selecting Records with Missing Data](#)
 - [Generating Other Nodes for Data Preparation](#)
-

Handling Outliers and Extreme Values

The audit report lists number of outliers and extremes is listed for each field based on the detection options specified in the Data Audit node. See the topic [Data Audit Quality Tab](#) for more information. You can choose to coerce, discard, or nullify these values for specific fields as appropriate, and then generate a SuperNode to apply the transformations.

1. In the Action column, specify handling for outliers and extremes for specific fields as desired.

The following actions are available for handling outliers and extremes:

- **Coerce.** Replaces outliers and extreme values with the nearest value that would not be considered extreme. For example if an outlier is defined to be anything above or below three standard deviations, then all outliers would be replaced with the highest or lowest value within this range.
 - **Discard.** Discards records with outlying or extreme values for the specified field.
 - **Nullify.** Replaces outliers and extremes with the null or system-missing value.
 - **Coerce outliers / discard extremes.** Discards extreme values only.
 - **Coerce outliers / nullify extremes.** Nullifies extreme values only.
2. To generate the SuperNode, from the menus choose:
Generate > Outlier & Extreme SuperNode

The Outlier SuperNode dialog box is displayed.

3. Select All fields or Selected fields only, and then click OK to add the generated SuperNode to the stream canvas.
4. Attach the SuperNode to the stream to apply the transformations.

Optionally, you can edit the SuperNode and zoom in to browse or make changes. Within the SuperNode, values are discarded, coerced, or nullified using a series of Select and/or Filler nodes as appropriate.

Related information

- [Data Audit node](#)
 - [Data Audit Node Settings Tab](#)
 - [Data Audit Quality Tab](#)
 - [Data Audit Output Browser](#)
 - [Viewing and Generating Graphs](#)
 - [Display Statistics](#)
 - [Data Audit Browser Quality Tab](#)
 - [Imputing Missing Values](#)
 - [Filtering Fields with Missing Data](#)
 - [Selecting Records with Missing Data](#)
 - [Generating Other Nodes for Data Preparation](#)
-

Filtering Fields with Missing Data

From the Data Audit browser, you can create a new Filter node based on the results of the Quality analysis by using the Generate Filter from Quality dialog box.

Mode. Select the desired operation for specified fields, either Include or Exclude.

- **Selected fields.** The Filter node will include/exclude the fields selected on the Quality tab. For example you could sort the table on the % Complete column, use Shift-click to select the least complete fields, and then generate a Filter node that excludes these fields.
- **Fields with quality percentage higher than.** The Filter node will include/exclude fields where the percentage of complete records is greater than the specified threshold. The default threshold is 50%.

Filtering Empty or Typeless Fields

Note that after data values have been instantiated, typeless or empty fields are excluded from the audit results and from most other output in IBM® SPSS® Modeler. These fields are ignored for purposes of modeling, but may bloat or clutter the data. If so, you can use the Data Audit browser to generate a Filter node from that removes these fields from the stream.

1. To make sure that all fields are included in the audit, including empty or typeless fields, click Clear All Values in the upstream source or Type node, or set Values to <Pass> for all fields.
2. In the Data Audit browser, sort on the % Complete column, select the fields that have zero valid values (or some other threshold) and use the Generate menu to produce a Filter node which can be added to the stream.

Related information

- [Data Audit node](#)
 - [Data Audit Node Settings Tab](#)
 - [Data Audit Quality Tab](#)
 - [Data Audit Output Browser](#)
 - [Viewing and Generating Graphs](#)
 - [Display Statistics](#)
 - [Data Audit Browser Quality Tab](#)
 - [Imputing Missing Values](#)
 - [Handling Outliers and Extreme Values](#)
 - [Selecting Records with Missing Data](#)
 - [Generating Other Nodes for Data Preparation](#)
-

Selecting Records with Missing Data

From the Data Audit browser, you can create a new Select node based on the results of the quality analysis.

1. In the Data Audit browser, choose the Quality tab.
2. From the menu, choose:
Generate...>Missing Values Select Node

The Generate Select Node dialog box is displayed.

Select when record is. Specify whether records should be kept when they are Valid or Invalid.

Look for invalid values in. Specify where to check for invalid values.

- **All fields.** The Select node will check all fields for invalid values.
- **Fields selected in table.** The Select node will check only the fields currently selected in the Quality output table.
- **Fields with quality percentage higher than.** The Select node will check fields where the percentage of complete records is greater than the specified threshold. The default threshold is 50%.

Consider a record invalid if an invalid value is found in. Specify the condition for identifying a record as invalid.

- **Any of the above fields.** The Select node will consider a record invalid if *any* of the fields specified above contains an invalid value for that record.
- **All of the above fields.** The Select node will consider a record invalid only if *all* of the fields specified above contain invalid values for that record.

Related information

- [Data Audit node](#)
 - [Data Audit Node Settings Tab](#)
 - [Data Audit Quality Tab](#)
 - [Data Audit Output Browser](#)
 - [Viewing and Generating Graphs](#)
 - [Display Statistics](#)
 - [Data Audit Browser Quality Tab](#)
 - [Imputing Missing Values](#)
 - [Handling Outliers and Extreme Values](#)
 - [Filtering Fields with Missing Data](#)
 - [Generating Other Nodes for Data Preparation](#)
-

Generating Other Nodes for Data Preparation

A variety of nodes used in data preparation can be generated directly from the Data Audit browser, including Reclassify, Binning, and Derive nodes. For example:

- You can derive a new field based on the values of *claimvalue* and *farmincome* by selecting both in the audit report and choosing Derive from the Generate menu. The new node is added to the stream canvas.
- Similarly, you may determine, based on audit results, that recoding *farmincome* into percentile-based bins provides a more focused analysis. To generate a Binning node, select the field row in the display and choose Binning from the Generate menu.

Once a node is generated and added to the stream canvas, you must attach it to the stream and open the node to specify options for the selected field(s).

Related information

- [Data Audit node](#)
 - [Data Audit Node Settings Tab](#)
 - [Data Audit Quality Tab](#)
 - [Data Audit Output Browser](#)
 - [Viewing and Generating Graphs](#)
 - [Display Statistics](#)
 - [Data Audit Browser Quality Tab](#)
 - [Imputing Missing Values](#)
 - [Handling Outliers and Extreme Values](#)
 - [Filtering Fields with Missing Data](#)
 - [Selecting Records with Missing Data](#)
-

Transform Node

Normalizing input fields is an important step before using traditional scoring techniques, such as regression, logistic regression, and discriminant analysis. These techniques carry assumptions about normal distributions of data that may not be true for many raw data files. One approach to dealing with real-world data is to apply transformations that move a raw data element toward a more normal distribution. In addition, normalized fields can easily be compared with each other—for example, income and age are on totally different scales in a raw data file but when normalized, the relative impact of each can be easily interpreted.

The Transform Node provides an output viewer that enables you to perform a rapid visual assessment of the best transformation to use. You can see at a glance whether variables are normally distributed and, if necessary, choose the transformation you want and apply it. You can pick multiple fields and perform one transformation per field.

After selecting the preferred transformations for the fields, you can generate Derive or Filler nodes that perform the transformations and attach these nodes to the stream. The Derive node creates new fields, while the Filler node transforms the existing ones. See the topic [Generating Graphs](#) for more information.

Transform node Fields tab

On the Fields tab, you can specify which fields of the data you want to use for viewing possible transformations and applying them. Only numeric fields can be transformed. Click the field selector button and select one or more numeric fields from the list displayed.

- [Transform Node Options Tab](#)
 - [Transform Node Output Tab](#)
 - [Transform Node Output Viewer](#)
-

Transform Node Options Tab

The Options tab enables you to specify the type of transformations you want to include. You can choose to include all available transformations or select transformations individually.

In the latter case, you can also enter a number to offset the data for the inverse and log transformations. Doing so is useful in situations where a large proportion of zeros in the data would bias the mean and standard deviation results.

For example, assume that you have a field named *BALANCE* that has some zero values in it, and you want to use the inverse transformation on it. To avoid undesired bias, you would select Inverse (1/x) and enter 1 in the Use a data offset field. (Note that this offset is not related to that performed by the @OFFSET sequence function in IBM® SPSS® Modeler.)

All formulas. Indicates that all available transformations should be calculated and shown in the output.

Select formulas. Enables you to select the different transformations to be calculated and shown in the output.

- **Inverse (1/x).** Indicates that the inverse transformation should be displayed in the output.

- **Log (log n).** Indicates that the \log_n transformation should be displayed in the output.
- **Log (log 10).** Indicates that the \log_{10} transformation should be displayed in the output.
- **Exponential.** Indicates that the exponential transformation (e^x) should be displayed in the output.
- **Square Root.** Indicates that the square root transformation should be displayed in the output.

Transform Node Output Tab

The Output tab lets you specify the output format and location of the output. You can also choose to display the results on the screen, or send them to one of the standard file types. See the topic [Output Node Output Tab](#) for more information.

Transform Node Output Viewer

The output viewer enables you to see the results of executing the Transform Node. The viewer is a powerful tool that displays multiple transformations per field in thumbnail views of the transformation, enabling you to compare fields quickly. You can use options on its File menu to save, export, or print the output. See the topic [Viewing output](#) for more information.

For each transformation (other than Selected Transform), a legend is displayed underneath in the format:

Mean (Standard deviation)

- [Generating Nodes for the Transformations](#)

Generating Nodes for the Transformations

The output viewer provides a useful starting point for your data preparation. For example, you might want to normalize the field *AGE* so that you can use a scoring technique (such as logistic regression or discriminant analysis) that assumes a normal distribution. Based upon the initial graphs and summary statistics, you might decide to transform the *AGE* field according to a particular distribution (for example, log). After selecting the preferred distribution, you can then generate a derive node with a standardized transformation to use for scoring.

You can generate the following field operations nodes from the output viewer:

- Derive
- Filler

A Derive node creates new fields with the desired transformations, while the Filler node transforms existing fields. The nodes are placed on the canvas in the form of a SuperNode.

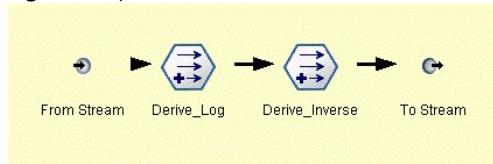
If you select the same transformation for different fields, a Derive or Filler node contains the formulas for that transformation type for all the fields to which that transformation applies. For example, assume that you have selected the fields and transformations, shown in the following table, to generate a Derive node.

Table 1. Example of Derive node generation

Field	Transformation
<i>AGE</i>	Current Distribution
<i>INCOME</i>	Log
<i>OPEN_BAL</i>	Inverse
<i>BALANCE</i>	Inverse

The following nodes are contained in the SuperNode:

Figure 1. SuperNode on canvas



In this example, the Derive_Log node has the log formula for the *INCOME* field, and the Derive_Inverse node has the inverse formulas for the *OPEN_BAL* and *BALANCE* fields.

To Generate a Node

1. For each field in the output viewer, select the desired transformation.
2. From the Generate menu, choose Derive Node or Filler Node as desired.
Doing so displays the Generate Derive Node or Generate Filler Node dialog box, as appropriate.

Choose Non-standardized transformation or Standardized transformation (z-score) as desired. The second option applies a z score to the transformation; z scores represent values as a function of distance from the mean of the variable in standard deviations. For example, if you apply the log transformation to the AGE field and choose a standardized transformation, the final equation for the generated node will be:

$(\log(\text{AGE}) - \text{Mean}) / \text{SD}$

Once a node is generated and appears on the stream canvas:

1. Attach it to the stream.
2. For a SuperNode, optionally double-click the node to view its contents.
3. Optionally double-click a Derive or Filler node to modify options for the selected field(s).

- [Generating Graphs](#)
-

Generating Graphs

You can generate full-size histogram output from a thumbnail histogram in the output viewer.

To Generate a Graph

1. Double-click a thumbnail graph in the output viewer.
or
2. Select a thumbnail graph in the output viewer.

2. From the Generate menu, choose Graph output.

Doing so displays the histogram with a normal distribution curve overlaid. This enables you to compare how closely each available transformation matches a normal distribution.

Note: You can only generate a graph if the Transform node that created the output is connected to the stream.

- [Other Operations](#)
-

Other Operations

From the output viewer, you can also:

- Sort the output grid by the Field column.
 - Export the output to an HTML file. See the topic [Exporting Output](#) for more information.
-

Statistics Node

The Statistics node gives you basic summary information about numeric fields. You can get summary statistics for individual fields and correlations between fields.

- [Statistics Node Settings Tab](#)
 - [Statistics Output Browser](#)
-

Statistics Node Settings Tab

Examine. Select the field or fields for which you want individual summary statistics. You can select multiple fields.

Statistics. Select the statistics to report. Available options include Count, Mean, Sum, Min, Max, Range, Variance, Std Dev, Std Error of Mean, Median, and Mode.

Correlate. Select the field or fields that you want to correlate. You can select multiple fields. When correlation fields are selected, the correlation between each Examine field and the correlation field(s) will be listed in the output.

Correlation Settings. You can specify options for displaying the strength of correlations in the output. See the topic [Correlation Settings](#) for more information.

- [Correlation Settings](#)

Related information

- [Statistics Node](#)
- [Correlation Settings](#)
- [Statistics Output Browser](#)
- [Generating a Filter Node from Statistics](#)
- [Output Node Output Tab](#)

Correlation Settings

IBM® SPSS® Modeler can characterize correlations with descriptive labels to help highlight important relationships. The **correlation** measures the strength of relationship between two continuous (numeric range) fields. It takes values between -1.0 and 1.0. Values close to +1.0 indicate a strong positive association so that high values on one field are associated with high values on the other and low values are associated with low values. Values close to -1.0 indicate a strong negative association so that high values for one field are associated with low values for the other, and vice versa. Values close to 0.0 indicate a weak association, so that values for the two fields are more or less independent.

Using the Correlation Settings dialog box you can control display of correlation labels, change the thresholds that define the categories, and change the labels used for each range. Because the way you characterize correlation values depends greatly on the problem domain, you may want to customize the ranges and labels to fit your specific situation.

Show correlation strength labels in output. This option is selected by default. Deselect this option to omit the descriptive labels from the output.

Correlation Strength. There are two options for defining and labeling the strength of correlations:

- **Define correlation strength by importance (1-p).** Labels correlations based on importance, defined as 1 minus the significance, or 1 minus the probability that the difference in means could be explained by chance alone. The closer this value comes to 1, the greater the chance that the two fields are *not* independent—in other words, that some relationship exists between them. Labeling correlations based on importance is generally recommended over absolute value because it accounts for variability in the data—for example, a coefficient of 0.6 may be highly significant in one dataset and not significant at all in another. By default, importance values between 0.0 and 0.9 are labeled as *Weak*, those between 0.9 and 0.95 are labeled as *Medium*, and those between 0.95 and 1.0 are labeled as *Strong*.
- **Define correlation strength by absolute value.** Labels correlations based on the absolute value of the Pearson's correlation coefficient, which ranges between -1 and 1, as described above. The closer the absolute value of this measure comes to 1, the stronger the correlation. By default, correlations between 0.0 and 0.3333 (in absolute value) are labeled as *Weak*, those between 0.3333 and 0.6666 are labeled as *Medium*, and those between 0.6666 and 1.0 are labeled as *Strong*. Note, however, that the significance of any given value is difficult to generalize from one dataset to another; for this reason, defining correlations based on probability rather than absolute value is recommended in most cases.

Related information

- [Statistics Node](#)
- [Statistics Node Settings Tab](#)
- [Statistics Output Browser](#)
- [Generating a Filter Node from Statistics](#)

Statistics Output Browser

The Statistics node output browser displays the results of the statistical analysis and enables you to perform operations, including selecting fields, and generating new nodes based on the selection, and saving and printing the results. The usual saving, exporting, and printing options are available from the File menu, and the usual editing options are available from the Edit menu. See the topic [Viewing output](#) for more information.

When you first browse Statistics output, the results are expanded. To hide results after viewing them, use the expander control to the left of the item to collapse the specific results you want to hide or click the Collapse All button to collapse all results. To see results again after collapsing them, use the expander control to the left of the item to show the results or click the Expand All button to show all results.

The output contains a section for each *Examine* field, containing a table of the requested statistics.

- **Count.** The number of records with valid values for the field.
- **Mean.** The average (mean) value for the field across all records.
- **Sum.** The sum of values for the field across all records.
- **Min.** The minimum value for the field.
- **Max.** The maximum value for the field.
- **Range.** The difference between the minimum and maximum values.
- **Variance.** A measure of the variability in the values of a field. It is calculated by taking the difference between each value and the overall mean, squaring it, summing across all of the values, and dividing by the number of records.
- **Standard Deviation.** Another measure of variability in the values of a field, calculated as the square root of the variance.
- **Standard Error of Mean.** A measure of the uncertainty in the estimate of the field's mean if the mean is assumed to apply to new data.
- **Median.** The "middle" value for the field; that is, the value that divides the upper half of the data from the lower half of the data (based on values of the field).
- **Mode.** The most common single value in the data.

Correlations. If you specified correlate fields, the output also contains a section listing the Pearson correlation between the Examine field and each correlate field, and optional descriptive labels for the correlation values. See the topic [Correlation Settings](#) for more information.

Generate menu. The Generate menu contains node generation operations.

- **Filter.** Generates a Filter node to filter out fields that are uncorrelated or weakly correlated with other fields. See the topic [Generating a Filter Node from Statistics](#) for more information.
- [Generating a Filter Node from Statistics](#)

Related information

- [Viewing output](#)
- [Statistics Node](#)
- [Statistics Node Settings Tab](#)
- [Correlation Settings](#)
- [Generating a Filter Node from Statistics](#)

Generating a Filter Node from Statistics

The Filter node generated from a Statistics output browser will filter fields based on their correlations with other fields. It works by sorting the correlations in order of absolute value, taking the largest correlations (according to the criterion set in the Generate Filter from Statistics dialog box), and creating a filter that passes all fields that appear in any of those large correlations.

Mode. Decide how to select correlations. Include causes fields appearing in the specified correlations to be retained. Exclude causes the fields to be filtered.

Include/Exclude fields appearing in. Define the criterion for selecting correlations.

- **Top number of correlations.** Selects the specified number of correlations and includes/excludes fields that appear in any of those correlations.
- **Top percentage of correlations (%).** Selects the specified percentage ($n\%$) of correlations and includes/excludes fields that appear in any of those correlations.
- **Correlations greater than.** Selects correlations greater in absolute value than the specified threshold.

Related information

- [Statistics Node](#)
- [Statistics Node Settings Tab](#)
- [Correlation Settings](#)
- [Statistics Output Browser](#)

Means Node

The Means node compares the means between independent groups or between pairs of related fields to test whether a significant difference exists. For example, you can compare mean revenues before and after running a promotion or compare revenues from customers who didn't receive the promotion with those who did.

You can compare means in two different ways, depending on your data:

- **Between groups within a field.** To compare independent groups, select a test field and a grouping field. For example, you could exclude a sample of "holdout" customers when sending a promotion and compare mean revenues for the holdout group with all of the others. In this case, you would specify a single test field that indicates the revenue for each customer, with a flag or nominal field that indicates whether they received the offer. The samples are independent in the sense that each record is assigned to one group or another, and there is no way to link a specific member of one group to a specific member of another. You can also specify a nominal field with more than two values to compare the means for multiple groups. When executed, the node calculates a one-way ANOVA test on the selected fields. In cases where there are only two field groups, the one-way ANOVA results are essentially the same as an independent-samples *t* test. See the topic [Comparing Means for Independent Groups](#) for more information.
 - **Between pairs of fields.** When comparing means for two related fields, the groups must be paired in some way for the results to be meaningful. For example, you could compare the mean revenues from the same group of customers before and after running a promotion or compare usage rates for a service between husband-wife pairs to see if they are different. Each record contains two separate but related measures that can be compared meaningfully. When executed, the node calculates a paired-samples *t* test on each field pair selected. See the topic [Comparing Means Between Paired Fields](#) for more information.
- [Comparing Means for Independent Groups](#)
 - [Comparing Means Between Paired Fields](#)
 - [Means Node Options](#)
 - [Means Node Output Browser](#)

Comparing Means for Independent Groups

Select Between groups within a field in the Means node to compare the mean for two or more independent groups.

Grouping field. Select a numeric flag or nominal field with two or more distinct values that divides records into the groups you want to compare, such as those who received an offer versus those who did not. Regardless of the number of test fields, only one grouping field can be selected.

Test fields. Select one or more numeric fields that contain the measures you want to test. A separate test will be conducted for each field you select. For example, you could test the impact of a given promotion on usage, revenue, and churn.

Related information

- [Means Node](#)
- [Comparing Means Between Paired Fields](#)
- [Means Node Options](#)
- [Means Node Output Browser](#)
- [Means Output Comparing Groups within a Field](#)
- [Means Output Comparing Pairs of Fields](#)

Comparing Means Between Paired Fields

Select Between pairs of fields in the Means node to compare means between separate fields. The fields must be related in some way for the results to be meaningful, such as revenues before and after a promotion. Multiple field pairs can also be selected.

Field one. Select a numeric field that contains the first of the measures you want to compare. In a before-and-after study, this would be the "before" field.

Field two. Select the second field you want to compare.

Add. Adds the selected pair to the Test field pair(s) list.

Repeat field selections as needed to add multiple pairs to the list.

Correlation settings. Enables you to specify options for labeling the strength of correlations. See the topic [Correlation Settings](#) for more information.

Related information

- [Means Node](#)
- [Comparing Means for Independent Groups](#)
- [Means Node Options](#)
- [Means Node Output Browser](#)
- [Means Output Comparing Groups within a Field](#)
- [Means Output Comparing Pairs of Fields](#)

Means Node Options

The Options tab enables you to set the threshold p values used to label results as important, marginal, or unimportant. You can also edit the label for each ranking. Importance is measured on a percentage scale and can be broadly defined as 1 minus the probability of obtaining a result (such as the difference in means between two fields) as extreme as or more extreme than the observed result by chance alone. For example, a p value greater than 0.95 indicates less than a 5% chance that the result could be explained by chance alone.

Importance labels. You can edit the labels used to label each field pair or group in the output. The default labels are *important*, *marginal*, and *unimportant*.

Cutoff values. Specifies the threshold for each rank. Typically p values greater than 0.95 would rank as important, while those lower than 0.9 would be unimportant, but these thresholds can be adjusted as needed.

Note: Importance measures are available in a number of nodes. The specific computations depend on the node and on the type of target and input fields used, but the values can still be compared, since all are measured on a percentage scale.

Related information

- [Means Node](#)
- [Comparing Means for Independent Groups](#)
- [Comparing Means Between Paired Fields](#)
- [Means Node Output Browser](#)
- [Means Output Comparing Groups within a Field](#)
- [Means Output Comparing Pairs of Fields](#)

Means Node Output Browser

The Means output browser displays cross-tabulated data and enables you to perform standard operations including selecting and copying the table one row at a time, and sorting by any column, and saving and printing the table. See the topic [Viewing output](#) for more information.

The specific information in the table depends on the type of comparison (groups within a field or separate fields).

Sort by. Enables you to sort the output by a specific column. Click the up or down arrow to change the direction of the sort. Alternatively, you can click on any column heading to sort by that column. (To change the direction of the sort within the column, click again.)

View. You can choose Simple or Advanced to control the level of detail in the display. The advanced view includes all of the information from the simple view but with additional details provided.

- [Means Output Comparing Groups within a Field](#)
- [Means Output Comparing Pairs of Fields](#)

Related information

- [Means Node](#)
- [Comparing Means for Independent Groups](#)
- [Comparing Means Between Paired Fields](#)
- [Means Node Options](#)
- [Means Output Comparing Groups within a Field](#)
- [Means Output Comparing Pairs of Fields](#)

Means Output Comparing Groups within a Field

When comparing groups within a field, the name of the grouping field is displayed above the output table, and means and related statistics are reported separately for each group. The table includes a separate row for each test field.

The following columns are displayed:

- **Field.** Lists the names of the selected test fields.
- **Means by group.** Displays the mean for each category of the grouping field. For example, you might compare those who received a special offer (*New Promotion*) with those who didn't (*Standard*). In the advanced view, the standard deviation, standard error, and count are also displayed.
- **Importance.** Displays the importance value and label. See the topic [Means Node Options](#) for more information.

Advanced Output

In the advanced view, the following additional columns are displayed.

- **F-Test.** This test is based on the ratio of the variance between the groups and the variance within each group. If the means are the same for all groups, you would expect the *F* ratio to be close to 1 since both are estimates of the same population variance. The larger this ratio, the greater the variation between groups and the greater the chance that a significant difference exists.
- **df.** Displays the degrees of freedom.

Related information

- [Means Node](#)
- [Comparing Means for Independent Groups](#)
- [Comparing Means Between Paired Fields](#)
- [Means Node Options](#)
- [Means Node Output Browser](#)
- [Means Output Comparing Pairs of Fields](#)

Means Output Comparing Pairs of Fields

When comparing separate fields, the output table includes a row for each selected field pair.

- **Field One/Two.** Displays the name of the first and second field in each pair. In the advanced view, the standard deviation, standard error, and count are also displayed.
- **Mean One/Two.** Displays the mean for each field, respectively.
- **Correlation.** Measures the strength of relationship between two continuous (numeric range) fields. Values close to +1.0 indicate a strong positive association, and values close to -1.0 indicate a strong negative association. See the topic [Correlation Settings](#) for more information.
- **Mean Difference.** Displays the difference between the two field means.
- **Importance.** Displays the importance value and label. See the topic [Means Node Options](#) for more information.

Advanced Output

Advanced output adds the following columns:

95% Confidence Interval. Lower and upper boundaries of the range within which the true mean is likely to fall in 95% of all possible samples of this size from this population.

T-Test. The *t* statistic is obtained by dividing the mean difference by its standard error. The greater the absolute value of this statistic, the greater the probability that the means are not the same.

df. Displays the degrees of freedom for the statistic.

Related information

- [Means Node](#)
- [Comparing Means for Independent Groups](#)
- [Comparing Means Between Paired Fields](#)
- [Means Node Options](#)
- [Means Node Output Browser](#)
- [Means Output Comparing Groups within a Field](#)

Report Node

The Report node enables you to create formatted reports containing fixed text, as well as data and other expressions derived from the data. You specify the format of the report by using text templates to define the fixed text and the data output constructions. You can provide custom text formatting using HTML tags in the template and by setting options on the Output tab. Data values and other conditional output are included in the report using CLEM expressions in the template.

Alternatives to the Report Node

The Report node is most typically used to list records or cases output from a stream, such as all records meeting a certain condition. In this regard, it can be thought of as a less-structured alternative to the Table node.

- If you want a report that lists field information or anything else that is defined in the stream rather than the data itself (such as field definitions specified in a Type node), then a script can be used instead.
- To generate a report that includes multiple output objects (such as a collection of models, tables, and graphs generated by one or more streams) and that can be output in multiple formats (including text, HTML, and Microsoft Word/Office), an IBM® SPSS® Modeler project can be used.
- To produce a list of field names without using scripting, you can use a Table node preceded by a Sample node that discards all records. This produces a table with no rows, which can be transposed on export to produce a list of field names in a single column. (Select Transpose data on the Output tab in the Table node to do this.)
- [Report Node Template Tab](#)
- [Report Node Output Browser](#)

Related information

- [Overview of Output Nodes](#)
 - [Report Node Template Tab](#)
 - [Report Node Output Browser](#)
 - [Output Node Output Tab](#)
-

Report Node Template Tab

Creating a template. To define the contents of the report, you create a template on the Report node Template tab. The template consists of lines of text, each of which specifies something about the contents of the report, and some special tag lines used to indicate the scope of the content lines. Within each content line, CLEM expressions enclosed in square brackets ([]]) are evaluated before the line is sent to the report. There are three possible scopes for a line in the template:

Fixed. Lines that are not marked otherwise are considered fixed. Fixed lines are copied into the report only once, after any expressions that they contain are evaluated. For example, the line

```
This is my report, printed on [@TODAY]
```

would copy a single line to the report, containing the text and the current date.

Global (iterate ALL). Lines contained between the special tags #**ALL** and # are copied to the report once for each record of input data. CLEM expressions (enclosed in brackets) are evaluated based on the current record for each output line. For example, the lines

```
#ALL
For record [@INDEX], the value of AGE is [AGE]
#
```

would include one line for each record indicating the record number and age.

To generate a list of all records:

```
#ALL
[Age] [Sex] [Cholesterol] [BP]
#
```

Conditional (iterate WHERE). Lines contained between the special tags #**WHERE** <condition> and # are copied to the report once for each record where the specified condition is true. The condition is a CLEM expression. (In the **WHERE** condition, the brackets are optional.) For example, the lines

```
#WHERE [SEX = 'M']
Male at record no. [@INDEX] has age [AGE].
#
```

will write one line to the file for each record with a value of M for sex. The complete report will contain the fixed, global, and conditional lines defined by applying the template to the input data.

You can specify options for displaying or saving results using the Output tab, common to various types of output nodes. See the topic [Output Node Output Tab](#) for more information.

Outputting Data in HTML or XML Format

You can include HTML or XML tags directly in the template in order to write reports in either of these formats. For example, the following template produces an HTML table.

```
This report is written in HTML.
Only records where Age is above 60 are included.

<HTML>
<TABLE border="2">
```

```

<TR>
  <TD>Age</TD>
  <TD>BP</TD>
  <TD>Cholesterol</TD>
  <TD>Drug</TD>
</TR>

#WHERE Age > 60
<TR>
  <TD>[Age]</TD>
  <TD>[BP]</TD>
  <TD>[Cholesterol]</TD>
  <TD>[Drug]</TD>
</TR>
#
</TABLE>
</HTML>

```

Related information

- [Report Node](#)
 - [Report Node Output Browser](#)
 - [Output Node Output Tab](#)
-

Report Node Output Browser

The report browser shows you the contents of the generated report. The usual saving, exporting, and printing options are available from the File menu, and the usual editing options are available from the Edit menu. See the topic [Viewing output](#) for more information.

Related information

- [Viewing output](#)
 - [Report Node](#)
 - [Report Node Template Tab](#)
-

Set Globals Node

The Set Globals node scans the data and computes summary values that can be used in CLEM expressions. For example, you can use a Set Globals node to compute statistics for a field called *age* and then use the overall mean of *age* in CLEM expressions by inserting the function `@GLOBAL_MEAN(age)`.

- [Set Globals Node Settings Tab](#)

Related information

- [Overview of Output Nodes](#)
 - [Set Globals Node Settings Tab](#)
-

Set Globals Node Settings Tab

Globals to be created. Select the field or fields for which you want globals to be available. You can select multiple fields. For each field, specify the statistics to compute by making sure that the statistics you want are selected in the columns next to the field name.

- **MEAN.** The average (mean) value for the field across all records.
- **SUM.** The sum of values for the field across all records.
- **MIN.** The minimum value for the field.
- **MAX.** The maximum value for the field.
- **SDEV.** The standard deviation, which is a measure of variability in the values of a field and is calculated as the square root of the variance.

Default operation(s). The options selected here will be used when new fields are added to the Globals list above. To change the default set of statistics, select or deselect statistics as appropriate. You can also use the Apply button to apply the default operations to all fields in the list.

Note: Some operations are not applicable to non-numeric fields (for example, Sum for a date/time field). Operations that cannot be used with a selected field are disabled.

Clear all globals before executing. Select this option to remove all global values before calculating new values. If this option is not selected, newly calculated values replace older values, but globals that are not recalculated remain available, as well.

Display preview of globals created after execution. If you select this option, the Globals tab of the stream properties dialog box will appear after execution to display the calculated global values.

Related information

- [Set Globals Node](#)

Simulation Fitting Node

The Simulation Fitting node fits a set of candidate statistical distributions to each field in the data. The fit of each distribution to a field is assessed using a goodness of fit criterion. When a Simulation Fitting node is executed, a Simulation Generate node is built (or an existing node is updated). Each field is assigned its best fitting distribution. The Simulation Generate node can then be used to generate simulated data for each field.

Although the Simulation Fitting node is a terminal node, it does not add a model to the generated models palette, add an output or chart to the outputs tab, or export data.

Note: If the historical data is sparse (that is, there are lots of missing values), it may difficult for the fitting component to find enough valid values to fit distributions to the data. In cases where the data is sparse, before fitting you should either remove the sparse fields if they are not required, or impute the missing values. Using the options on the Quality tab of the Data Audit node you can view the number of complete records, identify which fields are sparse, and select an imputation method. If there are an insufficient number of records for distribution fitting, you can use a Balance node to increase the number of records.

Using a Simulation Fitting node to automatically create a Simulation Generate node

The first time the Simulation Fitting node is executed, a Simulation Generate node is created with an update link to the Simulation Fitting node. If the Simulation Fitting node is executed again, a new Simulation Generate node will be created only if the update link has been removed. A Simulation Fitting node can also be used to update a connected Simulation Generate node. The result depends on whether the same fields are present in both nodes, and if the fields are unlocked in the Simulation Generate node. See the topic [Simulation Generate Node](#) for more information.

A Simulation Fitting node can only have an update link to a Simulation Generate node. To define an update link to a Simulation Generate node, follow these steps:

1. Right-click the Simulation Fitting node.
2. From the menu, select Define Update Link.
3. Click the Simulation Generate node to which you want to define an update link.

To remove an update link between a Simulation Fitting node and a Simulation Generate node, right-click on the update link, and select Remove Link.

- [Distribution Fitting](#)
- [Simulation Fitting Node Settings Tab](#)

Related information

- [Overview of Output Nodes](#)
- [Distribution Fitting](#)
- [Simulation Fitting Node Settings Tab](#)

Distribution Fitting

A statistical distribution is the theoretical frequency of the occurrence of values that a variable can take. In the Simulation Fitting node, a set of theoretical statistical distributions is compared to each field of data. The distributions that are available for fitting are described in the topic [Distributions](#). The parameters of the theoretical distribution are adjusted to give the best fit to the data according to a measurement of the goodness of fit; either the Anderson-Darling criterion or the Kolmogorov-Smirnov criterion. The results of the distribution fitting by the Simulation Fitting node show which distributions were fitted, the best estimates of the parameters for each distribution, and how well each distribution fits the data. During distribution fitting, correlations between fields with numeric storage types, and contingencies between fields with a categorical distribution, are also calculated. The results of the distribution fitting are used to create a Simulation Generate node.

Before any distributions are fitted to your data, the first 1000 records are examined for missing values. If there are too many missing values, distribution fitting is not possible. If so, you must decide whether either of the following options are appropriate:

- Use an upstream node to remove records with missing values.
- Use an upstream node to impute values for missing values.

Distribution fitting does not exclude user-missing values. If your data have user-missing values and you want those values to be excluded from distribution fitting then you should set those values to system missing.

The role of a field is not taken into account when the distributions are fitted. For example, fields with the role Target are treated the same as fields with roles of Input, None, Both, Partition, Split, Frequency, and ID.

Fields are treated differently during distribution fitting according to their storage type and measurement level. The treatment of fields during distribution fitting is described in the following table.

Table 1. Distribution fitting according to storage type and measurement level of fields

Storage type			Measurement Level			
	Continuous	Categorical	Flag	Nominal	Ordinal	Typeless
String	Impossible		Categorical, dice and fixed distributions are fitted			
Integer						
Real						
Time	All distributions are fitted. Correlations and contingencies are calculated.		The categorical distribution is fitted. Correlations are not calculated.		Binomial, negative binomial and Poisson distributions are fitted, and correlations are calculated.	Field is ignored and not passed to the Simulation Generate node.
Date						
Timestamp						
Unknown			Appropriate storage type is determined from the data.			

Fields with the measurement level ordinal are treated like continuous fields and are included in the correlations table in the Simulation Generate node. If you want a distribution other than binomial, negative binomial or Poisson to be fitted to an ordinal field, you must change the measurement level of the field to continuous. If you have previously defined a label for each value of an ordinal field, and then change the measurement level to continuous, the labels will be lost.

Fields that have single values are not treated differently during distribution fitting to fields with multiple values. Fields with the storage type time, date, or timestamp are treated as numeric.

Fitting Distributions to Split Fields

If your data contains a split field, and you want distribution fitting to be carried out separately for each split, you must transform the data by using an upstream Restructure node. Using the Restructure node, generate a new field for each value of the split field. This restructured data can then be used for distribution fitting in the Simulation Fitting node.

Related information

- [Simulation Fitting Node](#)
 - [Simulation Fitting Node Settings Tab](#)
 - [Distributions](#)
-

Simulation Fitting Node Settings Tab

Source node name. You can generate the name of the generated (or updated) Simulation Generate node automatically by selecting Auto. The automatically generated name is the name specified in the Simulation Fitting node if a custom name has been specified (or Sim Gen if no custom name has been specified in the Simulation Fitting node). Select Custom to specify a custom name in the adjoining text field. Unless the text field is edited, the default custom name is Sim Gen.

Fitting Options With these options you can specify how the distributions are fitted to the fields and how the fit of the distributions is assessed.

- **Number of cases to sample.** This specifies the number of cases to use when fitting distributions to the fields in the data set. Select All cases to fit distributions to all of the records in the data. If your data set is very large, you might want to consider limiting the number of

cases that are used for distribution fitting. Select Limit to first N cases to use only the first N cases. Click the arrows to specify the number of cases to use. Alternatively, you can use an upstream node to take a random sample of records for distribution fitting.

- **Goodness of fit criteria (continuous fields only).** For continuous fields, select either the Anderson-Darling test or the Kolmogorov-Smirnov test of goodness of fit to rank distributions when fitting distributions to the fields. The Anderson-Darling test is selected by default and is especially recommended when you want to ensure the best possible fit in the tail regions. Both statistics are calculated for every candidate distribution, but only the selected statistic is used to order the distributions and determine the best fitting distribution.
- **Bins (empirical distribution only).** For continuous fields, the Empirical distribution is the cumulative distribution function of the historical data. It is the probability of each value, or range of values, and is derived directly from the data. You can specify the number of bins that are used for calculating the Empirical distribution for continuous fields by clicking the arrows. The default is 100 and the maximum is 1000.
- **Weight field (optional).** If your data set contains a weight field, click the field picker icon and select the weight field from the list. The weight field is then excluded from the distribution fitting process. The list shows all fields in the data set that have the measurement level continuous. You can select only one weight field.

Related information

- [Simulation Fitting Node](#)
 - [Distribution Fitting](#)
-

Simulation Evaluation Node

The Simulation Evaluation node is a terminal node that evaluates a specified field, provides a distribution of the field, and produces charts of distributions and correlations. This node is primarily used to evaluate continuous fields. It therefore complements the evaluation chart, which is generated by an Evaluation node and is useful for evaluating discrete fields. Another difference is that the Simulation Evaluation node evaluates a single prediction across several iterations, whereas the Evaluation node evaluates multiple predictions each with a single iteration. Iterations are generated when more than one value is specified for a distribution parameter in the Simulation Generate node. See the topic [Iterations](#) for more information.

The Simulation Evaluation node is designed to be used with data that was obtained from the Simulation Fitting and Simulation Generate nodes. The node can, however, be used with any other node. Any number of processing steps can be placed between the Simulation Generate node and the Simulation Evaluation node.

Important: The Simulation Evaluation node requires a minimum of 1000 records with valid values for the target field.

- [Simulation Evaluation Node Settings Tab](#)
- [Simulation Evaluation node output](#)

Related information

- [Overview of Output Nodes](#)
 - [Simulation Evaluation Node Settings Tab](#)
 - [Simulation Evaluation node output](#)
 - [Navigation Panel](#)
 - [Chart output](#)
 - [Chart Options](#)
-

Simulation Evaluation Node Settings Tab

On the Settings tab of the Simulation Evaluation node, you can specify the role of each field in your data set, and customize the output that is generated by the simulation.

Select an item. Enables you to switch between the three views of the Simulation Evaluation node: Fields, Density Functions, and Outputs.

Fields view

Target field. This is a required field. Click the arrow to select the target field of your data set from the drop-down list. The selected field can have either continuous, ordinal or nominal measurement level, but cannot have date or an unspecified measurement level.

Iteration field (optional). If your data has an iteration field that indicates to which iteration each record in your data belongs, you must select it here. This means that each iteration will be evaluated separately. Only fields with continuous, ordinal or nominal measurement levels can be selected.

Input data is already sorted by iteration. Only enabled if an iteration field is specified in the Iteration field (optional) field. Only select this option if you are sure that your input data is already sorted by the iteration field specified in Iteration field (optional).

Maximum number of iterations to plot. Only enabled if an iteration field is specified in the Iteration field (optional) field. Click the arrows to specify the number of iterations to plot. Specifying this number avoids trying to plot too many iterations on a single chart, which would make the plot difficult to interpret. The lowest level that the maximum number of iterations can be set to is 2; the highest level is 50. The maximum number of iterations to plot is initially set to 10.

Input fields for correlation tornado. The correlation tornado chart is a bar chart that displays the correlation coefficients between the specified target and each of the specified inputs. Click the field picker icon to select the input fields to be included in the tornado chart, from a list of available simulated inputs. Only input fields with continuous and ordinal measurement levels can be selected. Nominal, typeless, and date input fields are not available in the list and cannot be selected.

Density Functions view

With the options in this view, you can customize the output for probability density functions and cumulative distribution functions for continuous targets, as well as bar charts of predicted values for categorical targets.

Density functions. Density functions are the primary means of probing the set of outcomes from your simulation.

- **Probability density function (PDF).** Select this option to produce a probability density function for the target field. The probability density function displays the distribution of the target values. You can use the probability density function to determine the probability that the target is within a particular region. For categorical targets (targets with a measurement level of nominal or ordinal) a bar chart is generated that displays the percentage of cases that fall in each category of the target.
- **Cumulative distribution function (CDF).** Select this option to produce a cumulative distribution function for the target field. The cumulative distribution function displays the probability that the value of the target is less than or equal to a specified value. It is only available for continuous targets.

Reference lines (continuous). These options are enabled if Probability density function (PDF), Cumulative distribution function (CDF), or both, are selected. With these options, you can add various fixed vertical reference lines to probability density functions and cumulative distribution functions.

- **Mean.** Select this option to add a reference line at the mean value of the target field.
- **Median.** Select this option to add a reference line at the median value of the target field.
- **Standard Deviations.** Select this option to add reference lines at plus and minus a specified number of standard deviations from the mean of the target field. Selecting this option enables the adjoining Number field. Click the arrows to specify the number of standard deviations. The minimum number of standard deviations is 1 and the maximum is 10. The number of standard deviations is initially set to 3.
- **Percentiles.** Select this option to add reference lines at two percentile values of the distribution of the target field. Selecting this option enables the adjoining Bottom and Top text fields. For example, entering a value of 90 in the Top text field would add a reference line at the 90th percentile of the target, which is the value below which 90% of the observations fall. Likewise, a value of 10 in the Bottom text field represents the tenth percentile of the target, which is the value below which 10% of the observations fall.
- **Custom reference lines.** Select this option to add reference lines at specified values along the horizontal axis. Selecting this option enables the adjoining Values table. Every time that you enter a valid number into the Values table, a new empty row is appended to the bottom of the table. A *valid* number is a number within the range of values of the target field

Note: When multiple density functions or distribution functions (from multiple iterations) are displayed on a single chart, reference lines (other than custom lines) are separately applied to each function.

Categorical target (PDF only). These options are only enabled if Probability density function (PDF) is selected.

- **Category values to report.** For models with categorical target fields, the result of the model is a set of predicted probabilities, one for each category, that the target value falls in each category. The category with the highest probability is taken to be the predicted category and used in generating the bar chart for the probability density function. Select Predicted category to generate the bar chart. Select Predicted probabilities to generate histograms of the distribution of predicted probabilities for each of the categories of the target field. You can also select Both to generate both types of chart.
- **Grouping for sensitivity analysis.** Simulations that include sensitivity analysis iterations generate an independent target field (or predicted target field from a model) for each iteration that is defined by the analysis. There is one iteration for each value of the distribution parameter that is being varied. When iterations are present, the bar chart of the predicted category for a categorical target field is displayed as a clustered bar chart that includes the results for all iterations. Select either Group categories together or Group iterations together.

Outputs view

Percentile values of target distributions. With these options, you can choose to create a table of percentile values of the target distributions, and specify the percentiles to display.

Create a table of percentile values. For continuous target fields, select this option to obtain a table of specified percentiles of the target distributions. Choose one of the following options to specify the percentiles:

- **Quartiles.** Quartiles are the 25th, 50th, and 75th percentiles of the target field distribution. The observations are divided into four groups of equal size.
- **Intervals.** If you want an equal number of groups other than four, select Intervals. Selecting this option enables the adjoining Number field. Click the arrows to specify the number of intervals. The minimum number of intervals is 2 and the maximum is 100. The number of intervals is initially set to 10.
- **Custom percentiles.** Select Custom percentiles to specify individual percentiles, for example, the 99th percentile. Selecting this option enables the adjoining Values table. Every time that you enter a valid number, between 1 and 100, into the Values table, a new empty row is

appended to the bottom of the table.

Related information

- [Simulation Evaluation Node](#)
 - [Simulation Evaluation node output](#)
 - [Navigation Panel](#)
 - [Chart output](#)
 - [Chart Options](#)
-

Simulation Evaluation node output

When the Simulation Evaluation node is executed, the output is added to the Output manager. The Simulation Evaluation output browser displays the results of executing the Simulation Evaluation node. The usual saving, exporting, and printing options are available from the File menu, and the usual editing options are available from the Edit menu. See the topic [Viewing output](#) for more information. The View menu is only enabled if one of the charts is selected. It is not enabled for the distribution table or information outputs. From the View menu you can select Edit Mode to change the layout and look of the chart, or Explore Mode to explore the data and values represented by the chart. The Static mode fixes the chart reference lines (and sliders) in their current positions so they cannot be moved. The Static mode is the only mode where you can copy, print, or export the chart with its reference lines. To select this mode, click Static Mode on the View menu.

The Simulation Evaluation output browser window consists of two panels. On the left of the window, there is a navigation panel that displays thumbnail representations of the charts that were generated when the Simulation Evaluation node was executed. When a thumbnail is selected, the chart output is displayed on the panel on the right of the window.

- [Navigation Panel](#)
- [Chart output](#)
- [Chart Options](#)

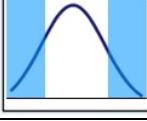
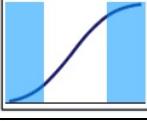
Related information

- [Simulation Evaluation Node](#)
 - [Simulation Evaluation Node Settings Tab](#)
 - [Navigation Panel](#)
 - [Chart output](#)
 - [Chart Options](#)
-

Navigation Panel

The navigation panel of the output browser contains thumbnails of the charts that are generated from a simulation. The thumbnails that are shown on the navigation panel depend on the measurement level of the target field, and on the options that are selected in the Simulation Evaluation node dialog box. Descriptions of the thumbnails are given in the following table.

Table 1. Navigation panel thumbnails

Thumbnail	Description	Comments
	Probability density function	This thumbnail is shown only if the measurement level of the target field is continuous, and Probability Density Function (PDF) is selected on the Density Functions view of the Simulation Evaluation node dialog box. If the measurement level of the target field is categorical, this thumbnail is not shown.
	Cumulative distribution function	This thumbnail is shown only if the measurement level of the target field is continuous, and Cumulative Distribution Function (CDF) is selected on the Density Functions view of the Simulation Evaluation node dialog box. If the measurement level of the target field is categorical, this thumbnail is not shown.
	Predicted category values	This thumbnail is shown only if the measurement level of the target field is categorical, Probability Density Function (PDF) is selected on the Density Functions view of the Simulation Evaluation node dialog box, and either Predicted category or Both is selected in the Category values to report area. If the measurement level of the target field is continuous, this thumbnail is not shown.

Thumbnail	Description	Comments
	Predicted category probabilities	This thumbnail is shown only if the measurement level of the target field is categorical, Probability Density Function (PDF) is selected on the Density Functions view of the Simulation Evaluation node dialog box, and either Predicted probabilities or Both is selected in the Category values to report area. If the measurement level of the target field is continuous, this thumbnail is not shown.
	Tornado charts	This thumbnail is shown only if one or more input fields are selected in the Input fields for correlation tornado field on the Fields view of the Simulation Evaluation node dialog box.
	Distribution table	This thumbnail is shown only if the measurement level of the target field is continuous, and Create a table of percentile values is selected on the Outputs view of the Simulation Evaluation node dialog box. The View menu is disabled for this chart. If the measurement level of the target field is categorical, this thumbnail is not shown.
	Information	This thumbnail is always shown. The View menu is disabled for this output.

Related information

- [Simulation Evaluation Node](#)
 - [Simulation Evaluation Node Settings Tab](#)
 - [Simulation Evaluation node output](#)
 - [Chart output](#)
 - [Chart Options](#)
-

Chart output

The types of output charts that are available depend on the measurement level of the target field, whether an iteration field is used, and the options that are selected in the Simulation Evaluation node dialog box. A number of the charts that are generated from a simulation have interactive features that you can use to customize the display. Interactive features are available by clicking Chart Options. All simulation charts are graphboard visualizations.

Probability density function charts for continuous targets. This chart shows both probability and frequency, with the probability scale on the left vertical axis and the frequency scale on the right vertical axis. The chart has two sliding vertical reference lines that divide the chart into separate regions. The table below the chart displays the percent of the distribution that is in each of the regions. If multiple density functions are displayed on the same chart (because of iterations), the table has a separate row for the probabilities that are associated with each density function, and an extra column that contains the iteration name and a color that is associated with each density function. The iterations are listed in the table in alphabetical order, according to the iteration label. If no iteration label is available, the iteration value is used instead. The table cannot be edited.

Each of the reference lines has a slider (inverted triangle) that you can use to easily move the line. Each slider has a label that indicates its current position. By default, the sliders are positioned at the 5th and 95th percentiles of the distribution. If there are multiple iterations, the sliders are positioned at the 5th and 95th percentiles of the first iteration that is listed in the table. You cannot move the lines to cross through each other.

A number of additional features are available by clicking Chart Options. In particular, you can explicitly set the positions of the sliders, add fixed reference lines, and change the chart view from a continuous curve to a histogram. See the topic [Chart Options](#) for more information. Right-click the chart to copy or export the chart.

Cumulative distribution function charts for continuous targets. This chart has the same two movable vertical reference lines and associated table that are described for the probability density function chart. The slider controls and table behave the same as the probability density function when there are multiple iterations. The same colors that are used to identify which density function belongs to each iteration are used for the distribution functions.

This chart also provides access to the Chart Options dialog box, which enables you to explicitly set the positions of the sliders, add fixed reference lines, and specify whether the cumulative distribution function is displayed as an increasing function (the default) or a decreasing function. See the topic [Chart Options](#) for more information. Right-click the chart to copy, export, or edit the chart. Selecting Edit opens the chart in a floating graphboard editor window.

Predicted category values chart for categorical targets. For categorical target fields, a bar chart displays the predicted values. The predicted values are displayed as the percent of the target field that is predicted to fall in each category. For categorical target fields with sensitivity analysis iterations, results for the predicted target category are displayed as a clustered bar chart that includes the results for all iterations. The chart is clustered by category or by iteration, depending on which option was selected in the Grouping for sensitivity analysis area in the Density Functions view of the Simulation Evaluation node dialog box. Right-click the chart to copy, export, or edit the chart. Selecting Edit opens the chart in a floating graphboard editor window.

Predicted category probabilities chart for categorical targets. For categorical target fields, a histogram displays the distribution of predicted probabilities for each of the categories of the target. For categorical target fields with sensitivity analysis iterations, the histograms are displayed by category or by iteration, depending on which option was selected in the Grouping for sensitivity analysis area in the Density Functions view of the Simulation Evaluation node dialog box. If the histograms are grouped by category, a drop-down list containing the iteration labels enables you to choose which iteration to display. You can also select the iteration to display by right-clicking the chart and selecting the iteration from the Iteration submenu. If the histograms are grouped by iteration, a drop-down list containing the category names enables you to choose which category to display. You can also select which category to display by right-clicking the chart and selecting the category from the Category submenu.

This chart is only available for a subset of models, and on the model nugget the option to generate all group probabilities must be selected. For example, on the Logistic model nugget, you must select Append all probabilities. The following model nuggets support this option:

- Logistic, SVM, Bayes, Neural Net and KNN
- Db2/ISW in-database mining models for logistic regression, decision trees and naïve Bayes

By default, the option to generate all group probabilities is not selected on these model nuggets.

Tornado charts. The tornado chart is a bar chart that shows the sensitivity of the target field to each of the specified inputs. The sensitivity is measured by the correlation of the target with each input. The title of the chart contains the name of the target field. Each bar on the chart represents the correlation between the target field and an input field. The simulated inputs that are included on the chart are the inputs that are selected in the Input fields for correlation tornado field on the Fields view of the Simulation Evaluation node dialog box. Each bar is labeled with the correlation value. Bars are ordered by the absolute value of the correlations, from largest value to smallest. If there are iterations, a separate chart is generated for each iteration. Each chart has a subtitle, which contains the name of the iteration.

Distribution table. This table contains the value of the target field, below which the specified percent of the observations fall. The table contains a row for each percentile value that is specified on the Outputs view of the Simulation Evaluation node dialog box. The percentile values can be quartiles, a different number of equally spaced percentiles, or individually specified percentiles. The distribution table contains a column for each iteration.

Information. This section gives an overall summary of the fields and records that are used in the evaluation. It also shows the input fields and record counts, which are broken down for each iteration.

Chart Options

In the Chart Options dialog box you can customize the display of activated charts of probability density functions and cumulative distribution functions that are generated from a simulation.

View. The View drop-down list only applies to the probability density function chart. You can use it to toggle the chart view from a continuous curve to a histogram. This feature is disabled when multiple density functions (from multiple iterations) are displayed on the same chart. When there are multiple density functions, the density functions can only be viewed as continuous curves.

Order. The Order drop-down list only applies to the cumulative distribution function chart. It specifies whether the cumulative distribution function is displayed as an ascending function (the default) or a descending function. When displayed as a descending function, the value of the function at a given point on the horizontal axis is the probability that the target field lies to the right of that point.

Slider positions. The Upper text field contains the current position of the right sliding reference line. The Lower text field contains the current position of the left sliding reference line. You can explicitly set the positions of the sliders by entering values in the Upper and Lower text fields. The value in the Lower text field must be strictly less than the value in the Upper text field. You can remove the left reference line by selecting -Infinity, effectively setting the position to negative infinity. This action disables the Lower text field. You can remove the right reference line by selecting Infinity, effectively setting its position to infinity. This action disables the Upper text field. You cannot remove both reference lines; selecting -Infinity disables the Infinity check box, and vice versa.

Reference lines. You can add various fixed vertical reference lines to probability density functions and cumulative distribution functions.

- **Mean.** You can add a reference line at the mean of the target field.
- **Median.** You can add a reference line at the median of the target field.
- **Standard Deviations.** You can add reference lines at plus and minus a specified number of standard deviations from the mean of the target field. You can enter the number of standard deviations to be used in the adjoining text field. The minimum number of standard deviations is 1, and the maximum is 10. The number of standard deviations is initially set to 3.
- **Percentiles.** You can add reference lines at one or two percentile values of the distribution for the target field by entering values into the Bottom and Top text fields. For example, a value of 95 in the Top text field represents the 95th percentile, which is the value below which 95% of the observations fall. Likewise, a value of 5 in the Bottom text field represents the fifth percentile, which is the value below which

5% of the observations fall. For the Bottom text field, the minimum percentile value is 0, and the maximum is 49. For the Top text field, the minimum percentile value is 50, and the maximum is 100.

- **Custom positions.** You can add reference lines at specified values along the horizontal axis. You can remove custom reference lines by deleting the entry from the grid.

When you click OK, the sliders, labels above the sliders, reference lines, and the table below the chart are updated to reflect the options that are selected in the Chart Options dialog box. Click Cancel to close the dialog box without making any changes. Reference lines can be removed by deselecting the associated choice in the Chart Options dialog and clicking OK.

Note: When multiple density functions or distribution functions are displayed on a single chart (because of results from sensitivity analysis iterations), reference lines (other than custom lines) are separately applied to each function. Only the reference lines for the first iteration are shown. The reference line labels include the iteration label. The iteration label is derived from upstream, typically from a Simulation Generate node. If no iteration label is available, the iteration value is used instead. The Mean, Median, Standard Deviations, and Percentiles options are disabled for cumulative distribution functions with multiple iterations.

Related information

- [Simulation Evaluation Node](#)
 - [Simulation Evaluation Node Settings Tab](#)
 - [Simulation Evaluation node output](#)
 - [Navigation Panel](#)
 - [Chart output](#)
-

Extension Output node

If Output to screen is selected on the Output tab of the Extension Output node dialog, on-screen output is displayed in an output browser window. The output is also added to the Output manager. The output browser window has its own set of menus that allow you to print or save the output, or export it to another format. The Edit menu only contains the Copy option. The Extension Output node's output browser has two tabs; the Text Output tab that displays text output, and the Graph Output tab that displays graphs and charts.

If Output to file is selected on the Output tab of the Extension Output node dialog, the output browser window is not displayed upon successful execution of the Extension Output node.

- [Extension Output node - Syntax tab](#)
 - [Extension Output node - Console Output tab](#)
 - [Extension Output node - Output tab](#)
 - [Extension Output Browser](#)
-

Extension Output node - Syntax tab

Select your type of syntax – R or Python for Spark. See the following sections for more information. When your syntax is ready, you can click Run to execute the Extension Output node. The output objects are added to the Output manager, or optionally to the file specified in the Filename field on the Output tab.

R Syntax

R Syntax. You can enter, or paste, custom R scripting syntax for data analysis into this field.

Convert flag fields. Specifies how flag fields are treated. There are two options: Strings to factor, Integers and Reals to double, and Logical values (True, False). If you select Logical values (True, False) the original values of the flag fields are lost. For example, if a field has values Male and Female, these are changed to True and False.

Convert missing values to the R 'not available' value (NA). When selected, any missing values are converted to the R NA value. The value NA is used by R to identify missing values. Some R functions that you use might have an argument that can be used to control how the function behaves when the data contain NA. For example, the function might allow you to choose to automatically exclude records that contain NA. If this option is not selected, any missing values are passed to R unchanged, and might cause errors when your R script is executed.

Convert date/time fields to R classes with special control for time zones. When selected, variables with date or datetime formats are converted to R date/time objects. You must select one of the following options:

- **R POSIXct.** Variables with date or datetime formats are converted to R POSIXct objects.
- **R POSIXlt (list).** Variables with date or datetime formats are converted to R POSIXlt objects.

Note: The POSIX formats are advanced options. Use these options only if your R script specifies that datetime fields are treated in ways that require these formats. The POSIX formats do not apply to variables with time formats.

Python Syntax

Python Syntax. You can enter, or paste, custom Python scripting syntax for data analysis into this field. For more information about Python for Spark, see [Python for Spark](#) and [Scripting with Python for Spark](#).

Extension Output node - Console Output tab

The Console Output tab contains any output that is received when the R script or Python for Spark script on the Syntax tab runs (for example, if using an R script, it shows output received from the R console when the R script in the R Syntax field on the Syntax tab is executed). This output might include R or Python error messages or warnings that are produced when the R or Python script is executed. The output can be used, primarily, to debug the script. The Console Output tab also contains the script from the R Syntax or Python Syntax field.

Every time the Extension Output script is executed, the content of the Console Output tab is overwritten with the output received from the R console or Python for Spark. The output cannot be edited.

Extension Output node - Output tab

Output name. Specifies the name of the output that is produced when the node is executed. When Auto is selected, the name of the output is automatically set to "R Output" or "Python Output" depending on the script type. Optionally, you can select Custom to specify a different name.

Output to screen. Select this option to generate and display the output in a new window. The output is also added to the Output manager.

Output to file. Select this option to save the output to a file. Doing so enables the Output Graph and Output File radio buttons.

Output Graph. Only enabled if Output to file is selected. Select this option to save any graphs that result from executing the Extension Output node to a file. Specify a filename to use for the generated output in the Filename field. Click the ellipses button (...) to choose a specific file and location. Specify the file type in the File type drop-down list. The following file types are available:

- Output object (.cou)
- HTML (.html)

Output Text. Only enabled if Output to file is selected. Select this option to save any text output that results from executing the Extension Output node to a file. Specify a filename to use for the generated output in the Filename field. Click the ellipses button (...) to choose a specific file and location. Specify the file type in the File type drop-down list. The following file types are available:

- HTML (.html)
- Output object (.cou)
- Text document (.txt)

Extension Output Browser

If Output to screen is selected on the Output tab of the Extension Output node dialog box, on-screen output is displayed in an output browser window. The output is also added to the Output manager. The output browser window has its own set of menus that allow you to print or save the output, or export it to another format. The Edit menu only contains the Copy option. The output browser of the Extension Output node has two tabs:

- The Text Output tab displays text output
- The Graph Output tab displays graphs and charts

If Output to file is selected on the Output tab of the Extension Output node dialog box instead of Output to screen, the output browser window is not displayed upon successful execution of the Extension Output node.

- [Extension Output Browser - Text Output tab](#)
- [Extension Output Browser - Graph Output tab](#)

Extension Output Browser - Text Output tab

The Text Output tab displays any text output that is generated when the R script or Python for Spark script on the Syntax tab of the Extension Output node is executed.

Note: R or Python for Spark error messages or warnings that result from executing your Extension Output script are always displayed on the Console Output tab of the Extension Output node.

Extension Output Browser - Graph Output tab

The Graph Output tab displays any graphs or charts that are generated when the R script or Python for Spark script on the Syntax tab of the Extension Output node is executed. For example, if your R script contains a call to the `R plot` function, the resulting graph is displayed on this tab.

KDE nodes

Kernel Density Estimation (KDE)© uses the Ball Tree or KD Tree algorithms for efficient queries, and walks the line between unsupervised learning, feature engineering, and data modeling. Neighbor-based approaches such as KDE are some of the most popular and useful density estimation techniques. KDE can be performed in any number of dimensions, though in practice high dimensionality can cause a degradation of performance. The KDE Modeling and KDE Simulation nodes in SPSS® Modeler expose the core features and commonly used parameters of the KDE library. The nodes are implemented in Python.¹

To use a KDE node, you must set up an upstream Type node. The KDE node will read input values from the Type node (or the Types tab of an upstream source node).

The KDE Modeling node is available on SPSS Modeler's Modeling tab and Python tab. The KDE Modeling node generates a model nugget, and the nugget's scored values are kernel density values from the input data.

The KDE Simulation node is available on the Output tab and the Python tab. The KDE Simulation node generates a KDE Gen source node that can create some records that have the same distribution as the input data. The KDE Gen node includes a Settings tab where you can specify how many records the node will create (default is 1) and generate a random seed.

For more information about KDE, including examples, see the KDE documentation available at <http://scikit-learn.org/stable/modules/density.html#kernel-density-estimation>.¹

¹ "User Guide." *Kernel Density Estimation*. Web. © 2007-2018, scikit-learn developers.

- [KDE Modeling node and KDE Simulation node Fields](#)
- [KDE nodes Build Options](#)
- [KDE Modeling node and KDE Simulation node Model Options](#)
- [KDE Modeling node and KDE Simulation node Fields](#)
- [KDE nodes Build Options](#)
- [KDE Modeling node and KDE Simulation node Model Options](#)

KDE Modeling node and KDE Simulation node Fields

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. This option uses the input settings from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign inputs, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the Inputs list on the right of the screen. The icons indicate the valid measurement levels for each field. To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Inputs. Select one or more fields as inputs for clustering. KDE can only deal with continuous fields.

KDE nodes Build Options

Use the Build Options tab to specify build options for the KDE nodes, including basic options for kernel density parameters and cluster labels, and advanced options such as tolerance, leaf size, and whether to use a breadth-first approach. For additional information about these options,

see the following online resources:

- [Kernel Density Estimation Python API Parameter Reference](#)¹
- [Kernel Density Estimation User Guide](#)²

Basic

Bandwidth. Specify the bandwidth of the kernel.

Kernel. Select the kernel to use. Available kernels for the KDE Modeling node are Gaussian, Tophat, Epanechnikov, Exponential, Linear, or Cosine. Available kernels for the KDE Simulation node are Gaussian or Tophat. For details about these available kernels, see the [Kernel Density Estimation User Guide](#).²

Algorithm. Select Auto, Ball Tree or KD Tree for the tree algorithm to use. For more information, see [Ball Tree](#)³ and [KD Tree](#).⁴

Metric. Select a distance metric. Available metrics are Euclidean, Braycurtis, Chebyshev, Canberra, Cityblock, Dice, Hamming, Infinity, Jaccard, L1, L2, Matching, Manhattan, P, Rogerstanimoto, Russellrao, Sokalmichener, Sokalsneath, Kulsinski, or Minkowski. If you select Minkowski, set the P Value as desired.

The metrics available in this drop-down will vary depending on which algorithm you choose. Also note that the normalization of the density output is correct only for the Euclidean distance metric.

Advanced

Absolute Tolerance. Specify the desired absolute tolerance of the result. A larger tolerance will generally result in faster run time. Default is 0.0.

Relative Tolerance. Specify the desired relative tolerance of the result. A larger tolerance will generally result in faster run time. Default is 1E-8.

Leaf Size. Specify the leaf size of the underlying tree. Default is 40. Changing the leaf size may significantly impact performance and required memory. For more information about the Ball Tree and KD Tree algorithms, see [Ball Tree](#)³ and [KD Tree](#).⁴

Breadth first. Select True if you want to use a breadth-first approach or False to use a depth-first approach.

The following table shows the relationship between the settings in the SPSS® Modeler KDE node dialogs and the Python KDE library parameters.

Table 1. Node properties mapped to Python library parameters

SPSS Modeler setting	Script name (property name)	KDE parameter
Inputs	inputs	
Bandwidth	bandwidth	bandwidth
Kernel	kernel	kernel
Algorithm	algorithm	algorithm
Metric	metric	metric
P Value	pValue	pValue
Absolute Tolerance	atol	atol
Relative Tolerance	rtol	Rtol
Leaf Size	leafSize	leafSize
Breadth first	breadthFirst	breadthFirst

¹ "API Reference." *sklearn.neighbors.KernelDensity*. Web. © 2007-2018, scikit-learn developers.

² "User Guide." *Kernel Density Estimation*. Web. © 2007-2018, scikit-learn developers.

³ "[Ball Tree](#)." *Five balltree construction algorithms*. © 1989, Omohundro, S.M., International Computer Science Institute Technical Report.

⁴ "[K-D Tree](#)." *Multidimensional binary search trees used for associative searching*. © 1975, Bentley, J.L., Communications of the ACM.

KDE Modeling node and KDE Simulation node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

IBM SPSS Statistics Helper Applications

If a compatible version of IBM® SPSS® Statistics is installed and licensed on your computer, you can configure IBM SPSS Modeler to process data with IBM SPSS Statistics functionality using the Statistics Transform, Statistics Model, Statistics Output, or Statistics Export nodes.

For information on product compatibility with the current version of IBM SPSS Modeler, see the corporate Support site at <http://www.ibm.com/support>.

To configure IBM SPSS Modeler to work with IBM SPSS Statistics and other applications, choose:

Tools > Options > Helper Applications

IBM SPSS Statistics Interactive. Enter the full path and name of the command (for example, C:\Program Files\IBM\SPSS\Statistics\<nn>\stats.exe) to be used when launching IBM SPSS Statistics directly on a data file produced by the Statistics Export node. See [Statistics Export Node](#) for more information.

Connection. If IBM SPSS Statistics Server is located on the same host as IBM SPSS Modeler Server, you can enable a connection between the two applications, which increases efficiency by leaving data on the server during analysis. Select Server to enable the Port option below. The default setting is Local.

Port. Specify the server port for IBM SPSS Statistics Server.

IBM SPSS Statistics Location Utility. To enable IBM SPSS Modeler to use the Statistics Transform, Statistics Model, and Statistics Output nodes, you must have a copy of IBM SPSS Statistics installed and licensed on the computer where the stream is run.

- If running IBM SPSS Modeler in local (stand-alone) mode, the licensed copy of IBM SPSS Statistics must be on the local computer. Click this button to specify the location of the local IBM SPSS Statistics installation you want to use for the licensing.
- In addition, if running in distributed mode against a remote IBM SPSS Modeler Server, you also need to run a utility at the IBM SPSS Modeler Server host to create the *statistics.ini* file, which indicates to IBM SPSS Statistics the installation path for IBM SPSS Modeler Server. To do this, from the command prompt, change to the IBM SPSS Modeler Server *bin* directory and, for Windows, run:

```
statisticsutility -location=<IBM SPSS Statistics installation path>/
```

Alternatively, for UNIX, run:

```
./statisticsutility -location=<IBM SPSS Statistics installation path>/bin
```

If you do not have a licensed copy of IBM SPSS Statistics on your local machine, you can still run the Statistics File node against a IBM SPSS Statistics server, but attempts to run other IBM SPSS Statistics nodes will display an error message.

Comments

If you have trouble running the IBM SPSS Statistics procedure nodes, consider the following tips:

- If field names used in IBM SPSS Modeler are longer than eight characters (for versions prior to IBM SPSS Statistics 12.0), longer than 64 characters (for IBM SPSS Statistics 12.0 and subsequent versions), or contain invalid characters, it is necessary to rename or truncate them before reading them into IBM SPSS Statistics. See [Renaming or Filtering Fields for IBM SPSS Statistics](#) for more information.
- If IBM SPSS Statistics was installed after IBM SPSS Modeler, you may need to specify the IBM SPSS Statistics location, as explained previously.

Export Nodes

- [Overview of Export Nodes](#)
- [Database Export Node](#)
- [Flat File Export Node](#)
- [Statistics Export Node](#)
- [Data Collection Export Node](#)
- [Analytic Server Export node](#)
- [IBM Cognos export node](#)
- [IBM Cognos TM1 Export Node](#)
- [SAS Export Node](#)
- [Excel Export Node](#)
- [Extension Export node](#)
- [XML Export Node](#)
- [JSON Export node](#)
- [Common Export node tabs](#)

Overview of Export Nodes

Export nodes provide a mechanism for exporting data in various formats to interface with your other software tools.

The following export nodes are available:

	The Database export node writes data to an ODBC-compliant relational data source. In order to write to an ODBC data source, the data source must exist and you must have write permission for it.
	The Flat File export node outputs data to a delimited text file. It is useful for exporting data that can be read by other analysis or spreadsheet software.
	The Statistics Export node outputs data in IBM® SPSS® Statistics .sav or .zsav format. The .sav or .zsav files can be read by IBM SPSS Statistics Base and other products. This is also the format used for cache files in IBM SPSS Modeler.
	The Data Collection export node outputs data in the format used by Data Collection market research software. A Data Collection Data Library must be installed to use this node.
	The IBM Cognos Export node exports data in a format that can be read by Cognos databases.
	The IBM Cognos TM1 Export node exports data in a format that can be read by Cognos TM1 databases.
	The SAS export node outputs data in SAS format, to be read into SAS or a SAS-compatible software package. Three SAS file formats are available: SAS for Windows/OS2, SAS for UNIX, or SAS Version 7/8.
	The Excel export node outputs data in the Microsoft Excel .xlsx file format. Optionally, you can choose to launch Excel automatically and open the exported file when the node is executed.
	The XML export node outputs data to a file in XML format. You can optionally create an XML source node to read the exported data back into the stream.
	The JSON export node outputs data in JSON format. See JSON Export node for more information.

Database Export Node

You can use Database nodes to write data to ODBC-compliant relational data sources, which are explained in the description of the Database source node. See the topic [Database source node](#) for more information.

Use the following general steps to write data to a database:

1. Install an ODBC driver and configure a data source to the database you want to use.
2. On the Database node Export tab, specify the data source and table you want to write to. You can create a new table or insert data into an existing one.
3. Specify additional options as needed.

These steps are described in more detail in the next several topics.

- [Database Node Export Tab](#)
- [Database Export Merge Options](#)
- [Database export schema options](#)
- [Database Export Index Options](#)
- [Database export advanced options](#)
- [Bulk loader programming](#)

Database Node Export Tab

Note: Some of the databases to which you can export might not support column names in tables that are more than 30 characters long. If an error message is displayed that mentions that your table has an incorrect column name, reduce the size of the name to fewer than 30 characters.

Data source. Shows the selected data source. Enter the name or select it from the drop-down list. If you don't see the desired database in the list, select Add new database connection and locate your database from the Database Connections dialog box. See [Adding a database connection](#) for more information.

Table name. Enter the name of the table to which you want to send the data. If you select the Insert into table option,

- You can select an existing table in the database by clicking the Select button.
- If you provide a table name that does not currently exist, a new table will be created with the specified name, and data will be inserted into it.

Create table. Select this option to create a new database table or to overwrite an existing database table.

Insert into table. Select this option to:

- Insert the data into new rows in an existing database table, or
- Insert the data into a table that doesn't exist. In this case, a new table will be created and data will be inserted as new rows in the newly created table.

Merge table. (Where available) Select this option to update selected database columns with values from corresponding source data fields. Selecting this option enables the Merge button, which displays a dialog from where you can map source data fields to database columns.

Drop existing table. Select this option to delete any existing table with the same name when creating a new table.

Delete existing rows. Select this option to delete existing rows from the table before exporting, when inserting into a table.

Note: If either of the two options above are selected, you will receive an Overwrite warning message when you execute the node. To suppress the warnings, deselect Warn when a node overwrites a database table on the Notifications tab of the User Options dialog box.

Default string size. Fields you have marked as typeless in an upstream Type node are written to the database as string fields. Specify the size of strings to be used for typeless fields.

Click Schema to open a dialog box where you can set various export options (for databases that support this feature), set SQL data types for your fields, and specify the primary key for purposes of database indexing. See [Database export schema options](#) for more information.

Click Indexes to specify options for indexing the exported table in order to improve database performance. See [Database Export Index Options](#) for more information.

Click Advanced to specify bulk loading and database commit options. See [Database export advanced options](#) for more information.

Quote table and column names. Select options used when sending a `CREATE TABLE` statement to the database. Tables or columns with spaces or nonstandard characters must be quoted.

- As needed. Select to allow IBM® SPSS® Modeler to automatically determine when quoting is needed on an individual basis.
- Always. Select to always enclose table and column names in quotes.
- Never. Select to disable the use of quotes.

Generate an import node for this data. Select to generate a Database source node for the data as exported to the specified data source and table. Upon execution, this node is added to the stream canvas.

Related information

- [Database Export Node](#)
 - [Database Export Merge Options](#)
 - [Database export schema options](#)
 - [Database Export Index Options](#)
-

Database Export Merge Options

This dialog enables you to map fields from the source data onto columns in the target database table. Where a source data field is mapped to a database column, the column value is replaced with the source data value when the stream is run. Unmapped source fields are left unchanged in the database.

Map Fields. This is where you specify the mapping between source data fields and database columns. Source data fields with the same name as columns in the database are mapped automatically.

- **Map.** Maps a source data field selected in the field list on the left of the button to a database column selected in the list on the right. You can map more than one field at a time, but the number of entries selected in both lists must be the same.
- **Unmap.** Removes the mapping for one or more selected database columns. This button is activated when you select a field or database column in the table on the right of the dialog.
- **Add.** Adds one or more source data fields selected in the field list on the left of the button to the list on the right ready for mapping. This button is activated when you select a field in the list on the left and no field with that name exists in the list on the right. Clicking this button maps the selected field to a new database column with the same name. The word <NEW> is displayed after the database column name to indicate that this is a new field.

Merge Rows. You use a key field, such as *Transaction ID*, to merge records with the same value in the key field. This is equivalent to a database "equi-join." Key values must be those of primary keys; that is, they must be unique, and cannot contain null values.

- **Possible keys.** Lists all fields found in all input data sources. Select one or more fields from this list and use the arrow button to add them as key fields for merging records. Any map field with a corresponding mapped database column is available as a key, except that fields added as new database columns (shown with <NEW> after the name) are not available.
- **Keys for merge.** Lists all fields used to merge records from all input data sources based on values of the key fields. To remove a key from the list, select one and use the arrow button to return it to the Possible Keys list. When more than one key field is selected, the option below is enabled.
- **Only include records which exist in database.** Performs a partial join; if the record is in the database and the stream, the mapped fields will be updated.
- **Add records to database.** Performs an outer join; all records in the stream will be merged (if the same record exists in the database) or added (if the record does not yet exist in the database).

To map a source data field to a new database column

1. Click the source field name in the list on the left, under Map Fields.
2. Click the Add button to complete the mapping.

To map a source data field to an existing database column

1. Click the source field name in the list on the left, under Map Fields.
2. Click the column name under Database Column on the right.
3. Click the Map button to complete the mapping.

To remove a mapping

1. In the list on the right, under Field, click the name of the field for which you want to remove the mapping.
2. Click the Unmap button.

To deselect a field in any of the lists

Hold down the CTRL key and click the field name.

Related information

- [Database Export Node](#)
 - [Database Node Export Tab](#)
 - [Database export schema options](#)
 - [Database Export Index Options](#)
-

Database export schema options

On the database export Schema dialog box, you can set options for database export (for databases that support these options), set SQL data types for your fields, specify which fields are primary keys, and customize the `CREATE TABLE` statement generated upon export.

The dialog box has several parts:

- The section at the top (if displayed) contains options for export to a database that supports these options. This section is not displayed if you are not connected to such a database.
- The text field in the center displays the template used to generate the `CREATE TABLE` command, which by default follows the format: `CREATE TABLE <table-name> <(table columns)>`
- The table in the lower portion enables you to specify the SQL data type for each field and to indicate which fields are primary keys as discussed below. The dialog box automatically generates the values of the `<table-name>` and `<(table columns)>` parameters based on the specifications in the table.

Setting database export options

If this section is displayed, you can specify a number of settings for export to the database. The database types that support this feature are as follows.

- SQL Server Enterprise and Developer editions. See the topic [Options for SQL Server](#) for more information.
- Oracle Enterprise or Personal editions. See the topic [Options for Oracle](#) for more information.

Customizing CREATE TABLE statements

Using the text field portion of this dialog box, you can add extra database-specific options to the `CREATE TABLE` statement.

1. Select the Customize CREATE TABLE command check box to activate the text window.
2. Add any database-specific options to the statement. Be sure to retain the text `<table-name>` and `(<table-columns>)` parameters because these are substituted for the real table name and column definitions by IBM® SPSS® Modeler.

Setting SQL data types

By default, IBM SPSS Modeler enables the database server to assign SQL data types automatically. To override the automatic type for a field, find the row corresponding to the field and select the desired type from the drop-down list in the *Type* column of the schema table. You can use Shift-click to select more than one row.

For types that take a length, precision, or scale argument (`BINARY`, `VARBINARY`, `CHAR`, `VARCHAR`, `NUMERIC`, and `NUMBER`), you should specify a length rather than allow the database server to assign an automatic length. For example, specifying a sensible value, such as `VARCHAR(25)`, for length ensures that the storage type in IBM SPSS Modeler will be overwritten if that is your intention. To override the automatic assignment, select *Specify* from the *Type* drop-down list and replace the type definition with the desired SQL type definition statement.

The easiest way to do this is to first select the type that is closest to the desired type definition and then select *Specify* to edit that definition. For example, to set the SQL data type to `VARCHAR(25)`, first set the type to `VARCHAR(length)` from the *Type* drop-down list, and then select *Specify* and replace the text length with the value 25.

Primary keys

If one or more columns in the exported table must have a unique value or combination of values for every row, you can indicate this by selecting the Primary Key check box for each field that applies. Most databases will not allow the table to be modified in a manner that invalidates a primary key constraint and will automatically create an index over the primary key to help enforce this restriction. (Optionally, you can create indexes for other fields in the Indexes dialog box. See the topic [Database Export Index Options](#) for more information.)

- [Options for SQL Server](#)
- [Options for Oracle](#)

Related information

- [Database Export Node](#)
- [Database Node Export Tab](#)
- [Database Export Merge Options](#)
- [Database Export Index Options](#)

Options for SQL Server

Use compression. If selected, creates tables for export with compression.

Compression for. Choose the level of compression.

- Row. Enables row-level compression (for example, the equivalent of `CREATE TABLE MYTABLE (...) WITH (DATA_COMPRESSION = ROW)` ; in SQL).
- Page. Enables page-level compression (for example, `CREATE TABLE MYTABLE (...) WITH (DATA_COMPRESSION = PAGE)` ; in SQL).

Options for Oracle

Oracle settings - Basic option

Use compression. If selected, creates tables for export with compression.

Compression for. Choose the level of compression.

- Default. Enables default compression (for example, `CREATE TABLE MYTABLE (...) COMPRESS` ; in SQL). In this case, it has the same effect as the Basic option.
- Basic. Enables basic compression (for example, `CREATE TABLE MYTABLE (...) COMPRESS BASIC` ; in SQL).

Oracle settings - Advanced option

Use compression. If selected, creates tables for export with compression.

Compression for. Choose the level of compression.

- Default. Enables default compression (for example, `CREATE TABLE MYTABLE (...) COMPRESS`; in SQL). In this case, it has the same effect as the Basic option.
- Basic. Enables basic compression (for example, `CREATE TABLE MYTABLE (...) COMPRESS BASIC`; in SQL).
- OLTP. Enables OLTP compression (for example, `CREATE TABLE MYTABLE (...) COMPRESS FOR OLTP`; in SQL).
- Query Low/High. (Exadata servers only) Enables hybrid columnar compression for query (for example, `CREATE TABLE MYTABLE (...) COMPRESS FOR QUERY LOW`; or `CREATE TABLE MYTABLE (...) COMPRESS FOR QUERY HIGH`; in SQL). Compression for query is useful in data warehousing environments; `HIGH` provides a higher compression ratio than `LOW`.
- Archive Low/High. (Exadata servers only) Enables hybrid columnar compression for archive (for example, `CREATE TABLE MYTABLE (...) COMPRESS FOR ARCHIVE LOW`; or `CREATE TABLE MYTABLE (...) COMPRESS FOR ARCHIVE HIGH`; in SQL). Compression for archive is useful for compressing data that will be stored for long periods of time; `HIGH` provides a higher compression ratio than `LOW`.

Database Export Index Options

The Indexes dialog box enables you to create indexes on database tables exported from IBM® SPSS® Modeler. You can specify the field sets you want to include and customize the `CREATE INDEX` command, as needed.

The dialog box has two parts:

- The text field at the top displays a template that can be used to generate one or more `CREATE INDEX` commands, which by default follows the format:
`CREATE INDEX <index-name> ON <table-name>`
- The table in the lower portion of the dialog box enables you to add specifications for each index you want to create. For each index, specify the index name and the fields or columns to include. The dialog box automatically generates the values of the `<index-name>` and `<table-name>` parameters accordingly.

For example, the generated SQL for a single index on the fields `empid` and `deptid` might look like this:

```
CREATE INDEX MYTABLE_IDX1 ON MYTABLE(EMPID,DEPTID)
```

You can add multiple rows to create multiple indexes. A separate `CREATE INDEX` command is generated for each row.

Customizing the `CREATE INDEX` Command

Optionally, you can customize the `CREATE INDEX` command for all indexes or for a specific index only. This gives you the flexibility to accommodate specific database requirements or options and to apply customizations to all indexes or only specific ones, as needed.

- Select Customize `CREATE INDEX` command at the top of the dialog box to modify the template used for all indexes added subsequently. Note that changes will not automatically apply to indexes that have already been added to the table.
- Select one or more rows in the table and then click Update selected indexes at the top of the dialog box to apply the current customizations to all selected rows.
- Select the Customize check box in each row to modify the command template for that index only.

Note that the values of the `<index-name>` and `<table-name>` parameters are generated automatically by the dialog box based on the table specifications and cannot be edited directly.

BITMAP KEYWORD. If you are using an Oracle database, you can customize the template to create a bitmap index rather than a standard index, as follows:

```
CREATE BITMAP INDEX <index-name> ON <table-name>
```

Bitmap indexes may be useful for indexing columns with a small number of distinct values. The resulting SQL might look this:

```
CREATE BITMAP INDEX MYTABLE_IDX1 ON MYTABLE(COLOR)
```

UNIQUE keyword. Most databases support the `UNIQUE` keyword in the `CREATE INDEX` command. This enforces a uniqueness constraint similar to a primary key constraint on the underlying table.

```
CREATE UNIQUE INDEX <index-name> ON <table-name>
```

Note that for fields actually designated as primary keys, this specification is not necessary. Most databases will automatically create an index for any fields specified as primary key fields within the `CREATE TABLE` command, so explicitly creating indexes on these fields is not necessary. See the topic [Database export schema options](#) for more information.

FILLCODE keyword. Some physical parameters for the index can be fine-tuned. For example, SQL Server enables the user to trade off the index size (after initial creation) against the costs of maintenance as future changes are made to the table.

```
CREATE INDEX MYTABLE_IDX1 ON MYTABLE (EMPID,DEPTID) WITH FILLCODE=20
```

Other Comments

- If an index already exists with the specified name, index creation will fail. Any failures will initially be treated as warnings, allowing subsequent indexes to be created and then re-reported as an error in the message log after all indexes have been attempted.
- For best performance, indexes should be created after data has been loaded into the table. Indexes must contain at least one column.
- Before executing the node, you can preview the generated SQL in the message log.
- For temporary tables written to the database (that is, when node caching is enabled) the options to specify primary keys and indexes are not available. However the system may create indexes on the temporary table as appropriate, depending on how the data is used in downstream nodes. For example, if cached data is subsequently joined by a *DEPT* column, it would make sense to index the cached table on this column.

Indexes and Query Optimization

In some database management systems, once a database table has been created, loaded, and indexed, a further step is required before the optimizer is able to utilize the indexes to speed up query execution on the new table. For example, in Oracle, the cost-based query optimizer requires that a table be analyzed before its indexes can be used in query optimization. The internal ODBC properties file for Oracle (not user-visible) contains an option to make this happen, as follows:

```
# Defines SQL to be executed after a table and any associated indexes
# have been created and populated
table_analysis_sql, 'ANALYZE TABLE <table-name> COMPUTE STATISTICS'
```

This step is executed whenever a table is created in Oracle (regardless of whether primary keys or indexes are defined). If necessary, the ODBC properties file for additional databases can be customized in a similar way - contact Support for assistance.

Related information

- [Database Export Node](#)
 - [Database Node Export Tab](#)
 - [Database Export Merge Options](#)
 - [Database export schema options](#)
-

Database export advanced options

When you click the Advanced button from the Database export node dialog box, a new dialog box is displayed in which you can specify technical details for exporting results to a database.

Use batch commit. Select to turn off row-by-row commits to the database.

Batch size. Specifies the number of records to send to the database before committing to memory. Lowering this number provides greater data integrity at the cost of slower transfer speeds. You may want to fine-tune this number for optimal performance with your database.

Use bulk loading. Specifies a method for bulk loading data to the database directly from IBM® SPSS® Modeler. Some experimentation may be required to select which bulk load options are appropriate for a particular scenario.

- Via ODBC. Select to use the ODBC API to execute multiple-row inserts with greater efficiency than normal export to the database. Choose from row-wise or column-wise binding in the options below.
- Via external loader. Select to use a custom bulk loader program specific to your database. Selecting this option activates a variety of options below.

Advanced ODBC Options. These options are available only when Via ODBC is selected. Note that this functionality may not be supported by all ODBC drivers.

- Row-wise. Select row-wise binding to use the `SQLBulkOperations` call for loading data into the database. Row-wise binding typically improves speed compared to the use of parameterized inserts that insert data on a record-by-record basis.
- Column-wise. Select to use column-wise binding for loading data into the database. Column-wise binding improves performance by binding each database column (in a parameterized `INSERT` statement) to an array of *N* values. Executing the `INSERT` statement once causes *N* rows to be inserted into the database. This method can dramatically increase performance.

External Loader Options. When Via external loader is specified, a variety of options are displayed for exporting the dataset to a file and specifying and executing a custom loader program to load the data from that file into the database. IBM SPSS Modeler can interface with external loaders for many popular database systems. Several scripts have been included with the software and are available along with technical documentation under the scripts subdirectory. Note that in order to use this functionality, Python 2.7 must be installed on the same machine as IBM SPSS

Modeler or IBM SPSS Modeler Server, and the `python_exe_path` parameter must be set in the options.cfg file. See the topic [Bulk loader programming](#) for more information.

- Use delimiter. Specifies which delimiter character should be used in the exported file. Select Tab to delimit with tab and Space to delimit with spaces. Select Other to specify another character, such as a comma (,).
- Specify data file. Select to enter the path to use for the data file written during bulk loading. By default, a temporary file is created in the temp directory on the server.
- Specify loader program. Select to specify a bulk loading program. By default, the software searches the scripts subdirectory of the IBM SPSS Modeler installation for a Python script to execute for a given database. Several scripts have been included with the software and are available along with technical documentation under the scripts subdirectory.
- Generate log. Select to generate a log file to the specified directory. The log file contains error information and is useful if the bulk load operation fails.
- Check table size. Select to perform table checking that ensures that the increase in table size corresponds to the number of rows exported from IBM SPSS Modeler.
- Extra loader options. Specifies additional arguments to the loader program. Use double quotes for arguments containing spaces.

Double quotes are included in optional arguments by escaping with a backslash. For example, the option specified as `-comment "This is a \\"comment\\""` includes both the `-comment` flag and the comment itself rendered as `This is a "comment"`.

A single backslash can be included by escaping with another backslash. For example, the option specified as `-specialdir "C:\\\\Test Scripts\\\\"` includes the flag `-specialdir` and the directory rendered as `C:\\Test Scripts\\`.

Bulk loader programming

The Database export node includes options for bulk loading on the Advanced Options dialog box. Bulk loader programs can be used to load data from a text file into a database.

The option Use bulk loading - via external loader configures IBM® SPSS® Modeler to do three things:

- Create any required database tables.
- Export the data to a text file.
- Invoke a bulk loader program to load the data from this file into the database table.

Typically the bulk loader program is not the database load utility itself (for example, Oracle's sqlldr utility) but a small script or program which forms the correct arguments, creates any database-specific auxiliary files (such as a control file) and then invokes the database load utility. The information in the following sections will help you edit an existing bulk loader.

Alternatively you can write your own program for bulk loading. See the topic [Developing bulk loader programs](#) for more information. Note that this is not covered under a standard Technical Support agreement, and you should contact an IBM Services representative for assistance.

Scripts for bulk loading

IBM SPSS Modeler ships with a number of bulk loader programs for different databases that are implemented using Python scripts. When you run a stream containing a Database export node with the Via external loader option selected, IBM SPSS Modeler creates the database table (if required) via ODBC, exports the data to a temporary file on the host running IBM SPSS Modeler Server, then invokes the bulk load script. This script in turn executes utilities provided by the DBMS vendor to upload data from the temporary files to the database.

Notes:

- The IBM SPSS Modeler installation does not include a Python runtime interpreter, so a separate installation of Python is required. See the topic [Database export advanced options](#) for more information.
- Scripts are provided (in the \scripts folder of the IBM SPSS Modeler installation directory) for the databases listed in the following table.
- Currently, the Bulk loader scripts provided by IBM SPSS Modeler don't support LDAP.

Table 1. Bulk loader scripts provided

Database	Script name	Further information
IBM Db2	db2_loader.py	See the topic Bulk loading data to IBM Db2 databases for more information.
IBM Netezza	netezza_loader.py	See the topic Bulk loading data to IBM Netezza databases for more information.
Oracle	oracle_loader.py	See the topic Bulk loading data to Oracle databases for more information.
SQL Server	mssql_loader.py	See the topic Bulk loading data to SQL Server databases for more information.
Teradata	teradata_loader.py	See the topic Bulk loading data to Teradata databases for more information.

- [Bulk loading data to IBM Db2 databases](#)
- [Bulk loading data to IBM Netezza databases](#)
- [Bulk loading data to Oracle databases](#)
- [Bulk loading data to SQL Server databases](#)
- [Bulk loading data to Teradata databases](#)

- [Developing bulk loader programs](#)
- [Testing bulk loader programs](#)

Bulk loading data to IBM Db2 databases

The following points may help you to configure for bulk loading from IBM® SPSS® Modeler to an IBM Db2 database using the External Loader option in the DB Export Advanced Options dialog box.

Ensure that the Db2 command line processor (CLP) utility is installed

The script db2_loader.py invokes the Db2 LOAD command. Ensure that the command line processor (db2 on UNIX, db2cmd on Windows) is installed on the server on which db2_loader.py is to be executed (typically, the host running IBM SPSS Modeler Server).

Check whether the local database alias name is the same as the actual database name

The Db2 local database alias is the name used by Db2 client software to refer to a database in a local or remote Db2 instance. If the local database alias is different from the name of the remote database, supply the extra loader option:

`-alias <local_database_alias>`

For example, the remote database is named STARS on host GALAXY but the Db2 local database alias on the host running IBM SPSS Modeler Server is STARS_GALAXY. Use the extra loader option

`-alias STARS_GALAXY`

Non-ASCII character data encoding

If you are bulk loading data that is not in ASCII format, you should ensure that the codepage variable in the configuration section of db2_loader.py is correctly set up on your system.

Blank strings

Blank strings are exported to the database as NULL values.

Bulk loading data to IBM Netezza databases

The following points may help you to configure for bulk loading from IBM® SPSS® Modeler to an IBM Netezza database using the External Loader option in the DB Export Advanced Options dialog box.

Ensure that the Netezza nzload utility is installed

The script netezza_loader.py invokes the Netezza utility *nzload*. Ensure that *nzload* is installed and correctly configured on the server on which *netezza_loader.py* is to be executed.

Exporting non-ASCII data

If your export contains data that is not in ASCII format, you may need to add `-encoding UTF8` to the Extra loader options field in the DB Export Advanced Options dialog box. This should ensure that non-ASCII data is correctly uploaded.

Date, Time and Timestamp format data

In the stream properties, set the date format to DD-MM-YYYY and the time format to HH:MM:SS.

Blank strings

Blank strings are exported to the database as NULL values.

Different order of columns in stream and target table when inserting data into an existing table

If the order of columns in the stream is different from that in the target table, data values will be inserted into the wrong columns. Use a Field Reorder node to ensure that the order of columns in the stream matches the order in the target table. See the topic [Field Reorder Node](#) for more information.

Tracking nzload progress

When running IBM SPSS Modeler in local mode, add **-sts** to the Extra loader options field in the DB Export Advanced Options dialog box to see status messages every 10000 rows in the command window opened by the *nzload* utility.

Bulk loading data to Oracle databases

The following points may help you to configure for bulk loading from IBM® SPSS® Modeler to an Oracle database using the External Loader option in the DB Export Advanced Options dialog box.

Ensure that the Oracle sqldr utility is installed

The script *oracle_loader.py* invokes the Oracle utility *sqldr*. Note that *sqldr* is not automatically included in Oracle Client. Ensure that *sqldr* is installed on the server on which *oracle_loader.py* is to be executed.

Specify the database SID or service name

If you are exporting data to a non-local Oracle server or your local Oracle server has multiple databases, you will need to specify the following in the Extra loader options field in the DB Export Advanced Options dialog box to pass in the SID or service name:

-database <SID>

Editing the configuration section in *oracle_loader.py*

On UNIX (and optionally, on Windows) systems, edit the configuration section at the start of the *oracle_loader.py* script. Here, values for ORACLE_SID, NLS_LANG, TNS_ADMIN and ORACLE_HOME environment variables can be specified if appropriate, as well as the full path of the *sqldr* utility.

Date, Time and Timestamp format data

In the stream properties, you should normally set the date format to YYYY-MM-DD and the time format to HH:MM:SS.

If you need to use different date and time formats from the above, consult your Oracle documentation and edit the *oracle_loader.py* script file.

Non-ASCII character data encoding

If you are bulk loading data that is not in ASCII format, you should ensure that the environment variable NLS_LANG is correctly set up on your system. This is read by the Oracle loader utility *sqldr*. For example, the correct value for NLS_LANG for Shift-JIS on Windows is Japanese_Japan.JA16SJIS. For more details on NLS_LANG, check your Oracle documentation.

Blank strings

Blank strings are exported to the database as NULL values.

Bulk loading data to SQL Server databases

The following points may help you to configure for bulk loading from IBM® SPSS® Modeler to a SQL Server database using the External Loader option in the DB Export Advanced Options dialog box.

Ensure that the SQL Server bcp.exe utility is installed

The script *mssql_loader.py* invokes the SQL Server utility *bcp.exe*. Ensure that *bcp.exe* is installed on the server where *mssql_loader.py* is to be executed.

Using spaces as delimiter does not work

Avoid choosing space as the delimiter in the DB Export Advanced Options dialog box.

Check table size option recommended

We recommend that you enable the Check table size option in the DB Export Advanced Options dialog box. Failures in the bulk load process are not always detected, and enabling this option performs an extra check that the correct number of rows has been loaded.

Blank strings

Blank strings are exported to the database as NULL values.

Specify the fully qualified SQL server named instance

There may be occasions when SPSS Modeler cannot access SQL Server due to an unqualified hostname and displays the following error:

Error encountered while executing external bulk loader. The log file may provide more details.

To correct this error, add the following string, including the double-quotes, to the Extra loader options field:

"-S mhreboot.spss.com\SQLEXPRESS"

Bulk loading data to Teradata databases

The following points may help you to configure for bulk loading from IBM® SPSS® Modeler to a Teradata database using the External Loader option in the DB Export Advanced Options dialog box.

Ensure that the Teradata fastload utility is installed

The script *teradata_loader.py* invokes the Teradata utility *fastload*. Ensure that *fastload* is installed and correctly configured on the server on which *teradata_loader.py* is to be run.

Data can be bulk loaded only to empty tables

You can only use empty tables as targets for a bulk load. If a target table contains any data prior to the bulk load, the operation will fail.

Date, Time and Timestamp format data

In the stream properties, set the date format to YYYY-MM-DD and the time format to HH:MM:SS.

Blank strings

Blank strings are exported to the database as NULL values.

Teradata process ID (tdpid)

By default, *fastload* exports data to the Teradata system with `tdpid=dbc`. Usually, there will be an entry in the HOSTS file which associates `dbccop1` with the IP address of the Teradata server. To use a different server, specify the following in the Extra loader options field in the DB Export Advanced Options dialog box to pass the `tdpid` of this server:

`-tdpid <id>`

Spaces in table and column names

If table or column names contain spaces, bulk load operation will fail. If possible, rename the table or column names to remove spaces.

Developing bulk loader programs

This topic explains how to develop a bulk loader program that can be run from IBM® SPSS® Modeler to load data from a text file into a database. Note that this is not covered under a standard Technical Support agreement, and you should contact an IBM Services representative for assistance.

Using Python to build bulk loader programs

By default, IBM SPSS Modeler searches for a default bulk loader program based on the database type. See [Table 1](#).

You can use the script *test_loader.py* to assist in developing batch loader programs. See the topic [Testing bulk loader programs](#) for more information.

Objects passed to the bulk loader program

IBM SPSS Modeler writes two files that are passed to the bulk loader program.

- **Data file.** This contains the data to be loaded, in text format.
- **Schema file.** This is an XML file that describes the names and types of the columns, and provides information on how the data file is formatted (for example, what character is used as the delimiter between fields).

In addition, IBM SPSS Modeler passes other information such as table name, user name and password as arguments when invoking the bulk load program.

Note: To signal successful completion to IBM SPSS Modeler, the bulk loader program should delete the schema file.

Arguments passed to the bulk loader program

The arguments passed to the program are listed in the following table.

Table 1. Arguments passed to bulk loader

Argument	Description
<code>schemafile</code>	Path of the schema file.
<code>data_file</code>	Path of the data file.
<code>servername</code>	Name of the DBMS server; may be blank.

Argument	Description
databasename	Name of the database within the DBMS server; may be blank.
username	User name to log into database.
password	Password to log into database.
tablename	Name of the table to load.
ownername	Name of the table owner (also known as schema name).
logfilename	Name of the logfile (if blank, no log file is created).
rowcount	Number of rows in the dataset.

Any options specified in the Extra loader options field on the DB Export Advanced Options dialog box are passed to the bulk loader program after these standard arguments.

Format of the data file

Data is written to the data file in text format, with each field being separated by a delimiter character that is specified on the DB Export Advanced Options dialog box. Here is an example of how a tab-delimited data file might appear.

```
48 F HIGH NORMAL 0.692623 0.055369 drugA
15 M NORMAL HIGH 0.678247 0.040851 drugY
37 M HIGH NORMAL 0.538192 0.069780 drugA
35 F HIGH HIGH 0.635680 0.068481 drugA
```

The file is written in the local encoding used by IBM SPSS Modeler Server (or IBM SPSS Modeler if not connected to IBM SPSS Modeler Server). Some formatting is controlled via the IBM SPSS Modeler stream settings.

Format of the schema file

The schema file is an XML file which describes the data file. Here is an example to accompany the preceding data file.

```
<?xml version="1.0" encoding="UTF-8"?>
<DBSCHEMA version="1.0">
    <table delimiter="\t" commit_every="10000" date_format="YYYY-MM-DD" time_format="HH:MM:SS"
append_existing="false" delete_datafile="false">
        <column name="Age" encoded_name="416765" type="integer"/>
        <column name="Sex" encoded_name="536578" type="char" size="1"/>
        <column name="BP" encoded_name="4250" type="char" size="6"/>
        <column name="Cholesterol" encoded_name="43686F6C65737465726F6C" type="char" size="6"/>
        <column name="Na" encoded_name="4E61" type="real"/>
        <column name="K" encoded_name="4B" type="real"/>
        <column name="Drug" encoded_name="44727567" type="char" size="5"/>
    </table>
</DBSCHEMA>
```

The following two tables list the attributes of the `<table>` and `<column>` elements of the schema file.

Table 2. Attributes of the `<table>` element

Attribute	Description
delimiter	The field delimiter character (TAB is represented as \t).
commit_every	The batch size interval (as on the DB Export Advanced Options dialog box).
date_format	The format used for representing dates.
time_format	The format used for representing times.
append_existing	true if the table to be loaded already contains data; false otherwise.
delete_datafile	true if the bulk load program should delete the data file on completion of loading.

Table 3. Attributes of the `<column>` element

Attribute	Description
name	The column name.
encoded_name	The column name converted to the same encoding as the data file and output as a series of two-digit hexadecimal numbers.
type	The data type of the column: one of integer , real , char , time , date , and datetime .
size	For the char data type, the maximum width of the column in characters.

Testing bulk loader programs

You can test bulk loading using a test script `test_loader.py` included in the `\scripts` folder of the IBM® SPSS® Modeler installation directory. Doing so is useful when trying to develop, debug, or troubleshoot bulk load programs or scripts for use with IBM SPSS Modeler.

To use the test script, proceed as follows.

1. Run the `test_loader.py` script to copy the schema and data files to the files `schema.xml` and `data.txt`, and create a Windows batch file (`test.bat`).
2. Edit the `test.bat` file to select the bulk loader program or script to test.
3. Run `test.bat` from a command shell to test the chosen bulk load program or script.

Note: Running `test.bat` does not actually load data into the database.

Flat File Export Node

The Flat File export node enables you to write data to a delimited text file. This is useful for exporting data that can be read by other analysis or spreadsheet software.

If your data contains geospatial information you can export it as a flat file and, if you generate a Variable File source node for use within the same stream, all storage, measurement, and geospatial metadata is retained in the new source node. However if you export the data and then import it in a different stream, you must take some extra steps to set the geospatial metadata in the new source node. For more information, see the topic [Variable File Node](#).

Note: You cannot write files in the old cache format, because IBM® SPSS® Modeler no longer uses that format for cache files. IBM SPSS Modeler cache files are now saved in IBM SPSS Statistics .sav format, which you can write using a Statistics export node. For more information, see the topic [Statistics Export Node](#).

- [Flat File Export Tab](#)
-

Flat File Export Tab

Export file. Specifies the name of the file. Enter a filename or click the file chooser button to browse to the file's location.

Write mode. If Overwrite is selected, any existing data in the specified file will be overwritten. If Append is selected, output will be added to the end of the existing file, preserving any data it contains.

- Include field names. If this option is selected, field names will be written to the first line of the output file. This option is available only for the Overwrite write mode.

New line after each record. If this option is selected, each record will be written on a new line in the output file.

Field separator. Specifies the character to insert between field values in the generated text file. Options are Comma, Tab, Space, and Other. If you select Other, enter the desired delimiter character(s) in the text box.

Symbol quotes. Specifies the type of quoting to use for values of symbolic fields. Options are None (values are not quoted), Single ('), Double ("), and Other. If you select Other, enter the desired quoting character(s) in the text box.

Encoding. Specifies the text-encoding method used. You can choose between the system default, stream default, or UTF-8.

- The system default is specified in the Windows Control Panel or, if running in distributed mode, on the server computer.
- The stream default is specified in the Stream Properties dialog box.

Decimal symbol. Specifies how decimals should be represented in the data.

- Stream default. The decimal separator defined by the current stream's default setting will be used. This will normally be the decimal separator defined by the computer's locale settings.
- Period (.). The period character will be used as the decimal separator.
- Comma (,). The comma character will be used as the decimal separator.

Generate an import node for this data. Select this option to automatically generate a Variable File source node that will read the exported data file. See the topic [Variable File Node](#) for more information.

Statistics Export Node

The Statistics Export node enables you to export data in IBM® SPSS® Statistics .sav format. IBM SPSS Statistics .sav files can be read by IBM SPSS Statistics Base and other modules. This is also the format used for IBM SPSS Modeler cache files.

Mapping IBM SPSS Modeler field names to IBM SPSS Statistics variable names can sometimes cause errors because IBM SPSS Statistics variable names are limited to 64 characters and cannot include certain characters, such as spaces, dollar signs (\$), and dashes (-). There are two ways to adjust for these restrictions:

- You can rename fields conforming to IBM SPSS Statistics variable name requirements by clicking the Filter tab. See the topic [Renaming or Filtering Fields for IBM SPSS Statistics](#) for more information.
- Choose to export both field names and labels from IBM SPSS Modeler.

Note: IBM SPSS Modeler writes .sav files in Unicode UTF-8 format. IBM SPSS Statistics only supports files in Unicode UTF-8 format from release 16.0 onwards. To prevent the possibility of data corruption .sav files saved with Unicode encoding should not be used in releases of IBM SPSS Statistics prior to 16.0. For more information, see the IBM SPSS Statistics help.

Multiple response sets. Any multiple response sets defined in the stream will automatically be preserved when the file is exported. You can view and edit multiple response sets from any node with a Filter tab. See the topic [Editing Multiple Response Sets](#) for more information.

- [Statistics Export Node - Export Tab](#)
- [Renaming or Filtering Fields for IBM SPSS Statistics](#)
- [Statistics Export Node - Export Tab](#)
- [Renaming or Filtering Fields for IBM SPSS Statistics](#)

Related information

- [Statistics Export Node - Export Tab](#)
- [Statistics File Node](#)
- [Statistics Transform Node](#)
- [Statistics Output Node](#)
- [Statistics Model Node](#)
- [Overview of Output Nodes](#)

Statistics Export Node - Export Tab

Export file Specifies the name of the file. Enter a filename or click the file chooser button to browse to the file's location.

File type Select if the file is to be saved in normal .sav or compressed.zsav format.

Encrypt file with password To protect the file with a password, select this box; you are prompted to enter and confirm the Password in a separate dialog box.

Note: Password protected files can only be opened by SPSS® Modeler version 16 or greater, or by SPSS Statistics version 21 or greater.

Export field names Specifies a method of handling variable names and labels upon export from SPSS Modeler to an SPSS Statistics .sav or .zsav file.

- Names and variable labels Select to export both SPSS Modeler field names and field labels. Names are exported as SPSS Statistics variable names, while labels are exported as SPSS Statistics variable labels.
- Names as variable labels Select to use the SPSS Modeler field names as variable labels in SPSS Statistics. SPSS Modeler allows characters in field names that are invalid in SPSS Statistics variable names. To prevent possibly creating invalid SPSS Statistics names, select Names as variable labels instead, or use the Filter tab to adjust field names.

Launch Application If SPSS Statistics is installed on your computer, you can select this option to invoke the application directly on the saved data file. Options for launching the application must be specified in the Helper Applications dialog box. See the topic [IBM SPSS Statistics Helper Applications](#) for more information. To simply create an SPSS Statistics .sav or .zsav file without opening an external program, deselect this option.

Note: When running SPSS Modeler and SPSS Statistics together in Server (distributed) mode, writing the data out and launching a SPSS Statistics session does not automatically open a SPSS Statistics client showing the data set read into the active data set. The workaround is to manually open the data file in the SPSS Statistics client once it is launched.

Generate an import node for this data Select this option to automatically generate a Statistics File source node that will read the exported data file. See the topic [Statistics File Node](#) for more information.

Related information

- [Statistics Export Node](#)

Renaming or Filtering Fields for IBM SPSS Statistics

Before exporting or deploying data from IBM® SPSS® Modeler to external applications such as IBM SPSS Statistics, it may be necessary to rename or adjust field names. The Statistics Transform, Statistics Output, and Statistics Export dialog boxes contain a Filter tab to facilitate this process.

A basic description of Filter tab functionality is discussed elsewhere. See the topic [Setting filtering options](#) for more information.

To adjust field names to conform to IBM SPSS Statistics naming conventions:

1. On the Filter tab, click the Filter Options Menu toolbar button (the first one on the toolbar).
2. Select Rename For IBM SPSS Statistics.
3. On the Rename For IBM SPSS Statistics dialog, you can choose to replace invalid characters in filenames with either a Hash (#) character or an Underscore (_).

Rename multi response sets. Select this option if you want to adjust the names of multiple response sets, which can be imported into IBM SPSS Modeler using a Statistics File source node. They are used to record data that can have more than one value for each case, such as in survey responses.

Related information

- [Statistics Export Node](#)
 - [Statistics Output Node](#)
 - [Statistics Output Node - Syntax Tab](#)
 - [Statistics Output Node - Output Tab](#)
-

Data Collection Export Node

The Data Collection export node saves data in the format used by Data Collection market research software, based on the Data Collection Data Model. This format distinguishes case data—the actual responses to questions gathered during a survey—from the metadata that describes how the case data is collected and organized. Metadata consists of information such as question texts, variable names and descriptions, multiple-response sets, translations of the various texts, and definitions of the structure of the case data. See the topic [Data Collection Node](#) for more information.

Metadata file. Specifies the name of the questionnaire definition file (.mdd) where the exported metadata will be saved. A default questionnaire is created based on field type information. For example, a nominal (set) field could be represented as a single question with the field description used as the question text and a separate check box for each defined value.

Merge metadata. Specifies whether the metadata will overwrite existing versions or be merged with existing metadata. If the merge option is selected, a new version is created each time the stream is run. This makes it possible to track versions of a questionnaire as it undergoes changes. Each version can be regarded as a snapshot of the metadata used to collect a particular set of case data.

Enable system variables. Specifies whether system variables are included in the exported .mdd file. These include variables such as *Respondent.Serial*, *Respondent.Origin*, and *DataCollection.StartTime*.

Case data settings. Specifies the IBM® SPSS® Statistics data (.sav) file where case data is exported. Note that all the restrictions on variable and value names apply here, so for example you may need to switch to the Filter tab and use the "Rename for IBM SPSS Statistics" option on the Filter options menu to correct invalid characters in field names.

Generate an import node for this data. Select this option to automatically generate a Data Collection source node that will read the exported data file.

Multiple response sets. Any multiple response sets defined in the stream will automatically be preserved when the file is exported. You can view and edit multiple response sets from any node with a Filter tab. See the topic [Editing Multiple Response Sets](#) for more information.

Analytic Server Export node

The Analytic Server Export node enables you to write data from your analysis to an existing Analytic Server data source. This can, for example, be text files on Hadoop Distributed File System (HDFS) or a database.

Typically, a stream with an Analytic Server Export node also begins with Analytic Server Source nodes, and is submitted to the Analytic Server and executed on HDFS. Alternatively, a stream with "local" data sources can end with an Analytic Server Export node in order to upload relatively small datasets (no more than 100,000 records) for use with Analytic Server.

If you want to use your own Analytic Server connection instead of the default connection defined by your administrator, deselect Use default Analytic Server and select your connection.

Data source. Select a data source containing the data you wish to use. A data source contains the files and metadata associated with that source. Click Select to display a list of available data sources. See the topic [Selecting a data source](#) for more information.

If you need to create a new data source or edit an existing one, click Launch Data Source Editor....

Mode. Select Append to add to the existing data source, or Overwrite to replace the contents of the data source.

Generate an Import node for this data. Select to generate a source node for the data as exported to the specified data source. This node is added to the stream canvas.

Note that using multiple Analytic Server connections can be useful in controlling the flow of data. For example, when using the Analytic Server Source and Export nodes, you may want to use different Analytic Server connections in different branches of a stream so that when each branch runs it uses its own Analytic Server and no data will be pulled to the IBM® SPSS® Modeler Server. Note that if a branch contains more than one Analytic Server connection, the data will be pulled from the Analytic Servers to the IBM SPSS Modeler Server.

IBM Cognos export node

The IBM Cognos Export node enables you to export data from an IBM® SPSS® Modeler stream to Cognos Analytics, in UTF-8 format. In this way, Cognos can make use of transformed or scored data from IBM SPSS Modeler. For example, you could use Cognos Report Studio to create a report based on the exported data, including the predictions and confidence values. The report could then be saved on the Cognos server and distributed to Cognos users.

Note: You can export only relational data, not OLAP data.

To export data to Cognos, you need to specify the following:

- Cognos connection - the connection to the Cognos Analytics server
- ODBC connection - the connection to the Cognos data server that the Cognos server uses

Within the Cognos connection you specify a Cognos data source to use. This data source must use the same login as the ODBC data source.

You export the actual stream data to the data server, and the package metadata to the Cognos server.

As with any other export node, you can also use the Publish tab of the node dialog box to publish the stream for deployment using IBM SPSS Modeler Solution Publisher.

Note: The Cognos source node only supports Cognos CQM packages. DQM packages are not supported.

- [Cognos connection](#)
- [ODBC connection](#)

Cognos connection

This is where you specify the connection to the Cognos Analytics server that you want to use for the export. The procedure involves exporting the metadata to a new package on the Cognos server, while the stream data is exported to the Cognos data server.

Connection. Click the Edit button to display a dialog box where you can define the URL and other details of the Cognos server to which you want to export the data. If you are already logged on to a Cognos server through IBM® SPSS® Modeler, you can also edit the details of the current connection. See [Cognos connections](#) for more information.

Data source. The name of the Cognos data source (typically a database) to which you are exporting the data. The drop-down list shows all the Cognos data sources that you can access from the current connection. Click the Refresh button to update the list.

Folder. The path and name of the folder on the Cognos server where the export package is to be created.

Package name. The name of the package in the specified folder that is to contain the exported metadata. This must be a new package with a single query subject; you cannot export to an existing package.

Mode. Specifies how you want to perform the export:

- Publish package now. (default) Performs the export operation as soon as you click Run.
- Export action script. Creates an XML script that you can run later (for example, using Framework Manager) to perform the export. Type the path and file name for the script in the File field, or use the Edit button to specify the name and location of the script file.

Generate an import node for this data. Select to generate a source node for the data as exported to the specified data source and table. When you click Run, this node is added to the stream canvas.

ODBC connection

Here you specify the connection to the Cognos data server (that is, the database) to which the stream data is to be exported.

Note: You must ensure that the data source you specify here points to the same one specified on the Cognos connections panel. You must also ensure that the Cognos connection data source uses the same login as the ODBC data source.

Data source. Shows the selected data source. Enter the name or select it from the drop-down list. If you don't see the desired database in the list, select Add new database connection and locate your database from the Database Connections dialog box. See [Adding a database connection](#) for more information.

Table name. Enter the name of the table to which you want to send the data. If you select the Insert into table option, you can select an existing table in the database by clicking the Select button.

Create table. Select this option to create a new database table or to overwrite an existing database table.

Insert into table. Select this option to insert the data as new rows in an existing database table.

Merge table. (Where available) Select this option to update selected database columns with values from corresponding source data fields. Selecting this option enables the Merge button, which displays a dialog from where you can map source data fields to database columns.

Drop existing table. Select this option to delete any existing table with the same name when creating a new table.

Delete existing rows. Select this option to delete existing rows from the table before exporting, when inserting into a table.

Note: If either of the two options above are selected, you will receive an Overwrite warning message when you execute the node. To suppress the warnings, deselect Warn when a node overwrites a database table on the Notifications tab of the User Options dialog box.

Default string size. Fields you have marked as typeless in an upstream Type node are written to the database as string fields. Specify the size of strings to be used for typeless fields.

Click Schema to open a dialog box where you can set various export options (for databases that support this feature), set SQL data types for your fields, and specify the primary key for purposes of database indexing. See [Database export schema options](#) for more information.

Click Indexes to specify options for indexing the exported table in order to improve database performance. See [Database Export Index Options](#) for more information.

Click Advanced to specify bulk loading and database commit options. See [Database export advanced options](#) for more information.

Quote table and column names. Select options used when sending a **CREATE TABLE** statement to the database. Tables or columns with spaces or nonstandard characters must be quoted.

- As needed. Select to allow IBM® SPSS® Modeler to automatically determine when quoting is needed on an individual basis.
- Always. Select to always enclose table and column names in quotes.
- Never. Select to disable the use of quotes.

Generate an import node for this data. Select to generate a source node for the data as exported to the specified data source and table. When you click Run, this node is added to the stream canvas.

IBM Cognos TM1 Export Node

The IBM Cognos Export node enables you to export data from an SPSS® Modeler stream to Cognos TM1. In this way, Cognos Analytics can make use of transformed or scored data from SPSS Modeler.

Note: You can export only measures, not context dimension data; alternatively, you can add new elements to the cube.

To export data to Cognos Analytics, you need to specify the following:

- The connection to the Cognos TM1 server.
- The Cube into which the data will be exported.
- The mapping from the SPSS data names to the equivalent TM1 dimensions and measures.

Note: The TM1 user needs the following permissions: Write privilege of cubes, Read privilege of dimensions, and Write privilege of dimension elements. In addition, IBM Cognos TM1 10.2 Fix Pack 3, or later, is required before SPSS Modeler can import and export Cognos TM1 data.

Existing streams that were based on previous versions will still function.

Administrator credentials are not required for this node. But if you are still using the old legacy pre-17.1 TM1 node, administrator credentials are still required.

SPSS Modeler only supports working with Cognos TM1 via IntegratedSecurityMode 1, 4, and 5.

As with any other export node, you can also use the Publish tab of the node dialog box to publish the stream for deployment using IBM® SPSS Modeler Solution Publisher.

Note: Before you can use the TM1 Source or Export nodes in SPSS Modeler, you must verify some settings in the tm1s.cfg file; this is the TM1 server configuration file in the root directory of the TM1 server.

- HTTPPortNumber - set a valid port number; typically 1-65535. Note that this is not the port number you subsequently specify in the connection in the node; it is an internal port used by TM1 that is disabled by default. If necessary, contact your TM1 administrator to

- confirm the valid setting for this port.
- UseSSL - if you set this to *True*, HTTPS is used as the transport protocol. In this case you must import the TM1 certification to the SPSS Modeler Server JRE.
 - [Connecting to an IBM Cognos TM1 cube to export data](#)
 - [Mapping IBM Cognos TM1 data for export](#)
-

Connecting to an IBM Cognos TM1 cube to export data

To export data to an IBM Cognos TM1 database, on the Connection tab of the IBM Cognos TM1 dialog box, specify the server connection details and select the associated cube and data details.

Note: Only actual "null" values will be discarded when exporting data to TM1. Zero (0) values will be exported as a valid value. Also note that only fields with a storage type of *string* can be mapped to dimensions on the Mapping tab. Before exporting to TM1, you must use IBM® SPSS® Modeler client to convert non-string data types to string.

Connection Type. Select Admin Server or TM1 Server. Note that the Admin Server has been removed from [Planning Analytics on Cloud](#), so if you have old streams that connect to an old Admin Server, you can modify them to point to Planning Analytics on Cloud instead. If you select Admin Server here, then you must enter the URL for the server (the host name of the REST API) and the name of the server. If you select TM1 Server, proceed to the following sections.

TM1 Server URL. Type the URL of the administration host where the TM1 server you want to connect to is installed. The administration host is defined as a single URL for all TM1 servers. From this URL, all IBM Cognos TM1 servers installed and running on your environment can be discovered and accessed. Click Login. If you have not previously connected to this server you are prompted to enter your User name and Password; alternatively, you can search for previously entered login details that you have saved as a Stored Credential.

Select a TM1 cube to export Displays the name of the cubes within the TM1 server into which you can export data.

To choose the data to export, select the cube and click the right arrow to move the cube into the Export to cube field. When you have selected the cube, use the Mapping tab to map the TM1 dimensions and measures to the relevant SPSS fields or to a fixed value (*Select* operation).

Mapping IBM Cognos TM1 data for export

After you select the TM1 administration host and associated TM1 server and cube, use the Mapping tab of the IBM Cognos TM1 Export dialog box to map TM1 dimensions and measures to SPSS fields or set TM1 dimensions to a fixed value.

Note: Only fields with a storage type of *string* can be mapped to dimensions. Before exporting to TM1, you must use IBM® SPSS® Modeler client to convert non-string data types to string.

Fields Lists the data field names from the SPSS data file that are available for export.

TM1 Dimensions Shows the TM1 cube selected on the Connection tab, along with its regular dimensions, the measure dimension, and the elements of the selected measure dimension. Select the name of the TM1 dimension, or measure, to map to the SPSS data field.

The following options are available on the Mapping tab.

Select a measure dimension From the list of dimensions for the selected cube, select one to be the measure dimension.

When you select a dimension, except the measure dimension, and click *Select*, a dialog is displayed that shows the leaf elements of the selected dimension. You can only select leaf elements. Selected elements are labeled with an S.

Map Maps the selected SPSS data field to the selected TM1 dimension or measure (a regular dimension or a specific measure or element from the measure dimension). Mapped fields are labeled with an M.

Unmap Unmaps the selected SPSS data field from the selected TM1 dimension or measure. Note that you can only unmap a single mapping at a time. The unmapped SPSS data field is moved back to the left column.

Create New Creates a new measure in the TM1 measure dimension. A dialog is displayed in which you enter the new TM1 measure name. This option is only available for measure dimensions, not for regular dimensions.

For more information about TM1, see the IBM Cognos TM1 documentation at http://www-01.ibm.com/support/knowledgecenter/SS9RXT_10.2.2/com.ibm.swg.ba.cognos.ctm1.doc/welcome.html.

Related information

- [IBM Cognos TM1 Export Node](#)
- [Connecting to an IBM Cognos TM1 cube to export data](#)

SAS Export Node

This feature is available in SPSS® Modeler Professional and SPSS Modeler Premium.

The SAS export node enables you to write data in SAS format to be read into SAS or a SAS-compatible software package. You can export in three SAS file formats: SAS for Windows/OS2, SAS for UNIX, or SAS.

- [SAS Export Node Export Tab](#)

Related information

- [Overview of Output Nodes](#)
- [SAS Export Node Export Tab](#)

SAS Export Node Export Tab

This feature is available in SPSS® Modeler Professional and SPSS Modeler Premium.

Export file. Specify the name of the file. Enter a filename or click the file chooser button to browse to the file's location.

Export. Specify the export file format. Options are SAS for Windows/OS2, SAS for UNIX, or SAS Version 7/8/9.

Export field names. Select options for exporting field names and labels from IBM® SPSS Modeler for use with SAS.

- Names and variable labels. Select to export both IBM SPSS Modeler field names and field labels. Names are exported as SAS variable names, while labels are exported as SAS variable labels.
- Names as variable labels. Select to use the IBM SPSS Modeler field names as variable labels in SAS. IBM SPSS Modeler allows characters in field names that are invalid in SAS variable names. To prevent possibly creating invalid SAS names, select Names and variable labels instead.

Generate an import node for this data. Select this option to automatically generate a SAS source node that will read the exported data file. See the topic [SAS Source Node](#) for more information.

Note: The maximum allowed length of a string is 255 bytes. If a string is more than 255 bytes, it will be truncated when exporting.

Excel Export Node

The Excel export node outputs data in Microsoft Excel .xlsx format. Optionally, you can choose to automatically launch Excel and open the exported file when the node is executed.

- [Excel Node Export Tab](#)

Excel Node Export Tab

File name. Enter a filename or click the file chooser button to browse to the file's location. The default filename is `excelexport.xlsx`.

File type. The Excel .xlsx file format is supported.

Create new file. Creates a new Excel file.

Insert into existing file. Content is replaced beginning at the cell designated by the Start in cell field. Other cells in the spreadsheet are left with their original content.

Include field names. Specifies whether field names should be included in the first row of the worksheet.

Start in cell. The cell location used for the first export record (or first field name if Include field names is checked). Data are filled to the right and down from this initial cell.

Choose worksheet. Specifies the worksheet to which you want to export the data. You can identify the worksheet either by index or by name:

- **By index.** If you are creating a new file, specify a number from 0 to 9 to identify the worksheet to which you want to export, beginning with 0 for the first worksheet, 1 for the second worksheet, and so on. You can use values of 10 or higher only if a worksheet already exists at this position.
- **By name.** If you are creating a new file, specify the name used for the worksheet. If you are inserting into an existing file, the data are inserted into this worksheet if it exists, otherwise a new worksheet with this name is created.

Launch Excel. Specifies whether Excel is automatically launched on the exported file when the node is executed. Note that when running in distributed mode against IBM® SPSS® Modeler Server, the output is saved to the server file system, and Excel is launched on the Client with a copy of the exported file.

Generate an import node for this data. Select this option to automatically generate an Excel source node that will read the exported data file. See the topic [Excel Source node](#) for more information.

Related information

- [Excel Export Node](#)
-

Extension Export node

With the Extension Export node, you can run R or Python for Spark scripts to export data.

- [Extension Export node - Syntax tab](#)
 - [Extension Export node - Console Output tab](#)
-

Extension Export node - Syntax tab

Select your type of syntax – R or Python for Spark. See the following sections for more information. When your syntax is ready, you can click Run to execute the Extension Export node.

R Syntax

R Syntax. You can enter, or paste, custom R scripting syntax for data analysis into this field.

Convert flag fields. Specifies how flag fields are treated. There are two options: Strings to factor, Integers and Reals to double, and Logical values (True, False). If you select Logical values (True, False) the original values of the flag fields are lost. For example, if a field has values Male and Female, these are changed to True and False.

Convert missing values to the R 'not available' value (NA). When selected, any missing values are converted to the R NA value. The value NA is used by R to identify missing values. Some R functions that you use might have an argument that can be used to control how the function behaves when the data contain NA. For example, the function might allow you to choose to automatically exclude records that contain NA. If this option is not selected, any missing values are passed to R unchanged, and might cause errors when your R script is executed.

Convert date/time fields to R classes with special control for time zones. When selected, variables with date or datetime formats are converted to R date/time objects. You must select one of the following options:

- **R POSIXct.** Variables with date or datetime formats are converted to R POSIXct objects.
- **R POSIXlt (list).** Variables with date or datetime formats are converted to R POSIXlt objects.

Note: The POSIX formats are advanced options. Use these options only if your R script specifies that datetime fields are treated in ways that require these formats. The POSIX formats do not apply to variables with time formats.

Python Syntax

Python Syntax. You can enter, or paste, custom Python scripting syntax for data analysis into this field. For more information about Python for Spark, see [Python for Spark](#) and [Scripting with Python for Spark](#).

Extension Export node - Console Output tab

The Console Output tab contains any output that is received when the R script or Python for Spark script on the Syntax tab runs (for example, if using an R script, it shows output received from the R console when the R script in the R Syntax field on the Syntax tab is executed). This output

might include R or Python error messages or warnings that are produced when the R or Python script is executed. The output can be used, primarily, to debug the script. The Console Output tab also contains the script from the R Syntax or Python Syntax field.

Every time the Extension Export script is executed, the content of the Console Output tab is overwritten with the output received from the R console or Python for Spark. The output cannot be edited.

XML Export Node

The XML Export node enables you to output data in XML format, using UTF-8 encoding. You can optionally create an XML source node to read the exported data back into the stream.

XML export file. The full path and file name of the XML file to which you want to export the data.

Use XML schema. Select this check box if you want to use a schema or DTD to control the structure of the exported data. Doing so activates the Map button, described below.

If you do not use a schema or DTD, the following default structure is used for the exported data:

```
<records>
  <record>
    <fieldname1>value</fieldname1>
    <fieldname2>value</fieldname2>
    :
    <fieldnameN>value</fieldnameN>
  </record>
  <record>
    :
    :
  </record>
  :
  :
</records>
```

Spaces in a field name are replaced with underscores; for example, "My Field" becomes `<My_Field>`.

Map. If you have chosen to use an XML schema, this button opens a dialog where you can specify which part of the XML structure should be used to start each new record. See the topic [XML Mapping Records Options](#) for more information.

Mapped fields. Indicates the number of fields that have been mapped.

Generate an import node for this data. Select this option to automatically generate an XML source node that will read the exported data file back into the stream. See the topic [XML Source Node](#) for more information.

- [Writing XML Data](#)
- [XML Mapping Records Options](#)
- [XML Mapping Fields Options](#)
- [XML Mapping Preview](#)

Related information

- [Overview of Output Nodes](#)

Writing XML Data

When an XML element is specified, the field value is placed inside the element tag:

```
<element>value</element>
```

When an attribute is mapped, the field value is placed as the value for the attribute:

```
<element attribute="value">
```

If a field is mapped to an element above the `<records>` element, the field is written only once, and will be a constant for all records. The value for this element will come from the first record.

If a null value is to be written, it is done by specifying empty content. For elements, this is:

```
<element></element>
```

For attributes, it is:

```
<element attribute="">
```

XML Mapping Records Options

The Records tab enables you to specify which part of the XML structure to use to start each new record. In order to map correctly onto a schema, you need to specify the record delimiter.

XML structure. A hierarchical tree showing the structure of the XML schema specified on the previous screen.

Records (XPath expression). To set the record delimiter, select an element in the XML structure and click the right-arrow button. Each time this element is encountered in the source data, a new record is created in the output file.

Note: If you select the root element in the XML structure, only a single record can be written, and all other records are skipped.

XML Mapping Fields Options

The Fields tab is used to map fields in the data set to elements or attributes in the XML structure when a schema file is used.

Field names that match an element or attribute name are automatically mapped as long as the element or attribute name is unique. Thus if there is both an element and an attribute with the name `field1`, there is no automatic mapping. If there is only one item in the structure named `field1`, a field with that name in the stream is automatically mapped.

Fields. The list of fields in the model. Select one or more fields as the source part of the mapping. You can use the buttons at the bottom of the list to select all fields, or all fields with a particular measurement level.

XML structure. Select an element in the XML structure as the map target. To create the mapping, click Map. The mapping is then displayed. The number of fields that have been mapped in this way is displayed below this list.

To remove a mapping, select the item in the XML structure list and click Unmap.

Display attributes. Displays or hides the attributes, if any, of the XML elements in the XML structure.

XML Mapping Preview

On the Preview tab, click Update to see a preview of the XML that will be written.

If the mapping is incorrect, return to the Records or Fields tab to correct the errors and click Update again to see the result.

JSON Export node

Use the JSON export node to output data in JSON format, using UTF-8 encoding. Optionally, you can also create a JSON source node to read the exported data back into the stream.

When SPSS® Modeler writes data to a JSON export file, it performs the following translations.

Table 1. JSON data export translation

SPSS Modeler Data Storage	JSON Value
String	string
Integer	number(int)
Real	number(real)
Date	string
Time	string
Timestamp	string
List	Not supported. Fields of list will be excluded.
Missing values	null

JSON export file. The full path and file name of the JSON file where data will be exported.

JSON string format. Specify the format of the JSON string. Select Records if you want the JSON export node to output a collection of name and value pairs. Or select Values if you only want to export values (no names).

JSON string format. Specify the format of the JSON string. Select Records the JSON export node will output a collection of name and value pairs. Or select Values if you only want to export values (no names).

Generate an import node for this data. Select this option to automatically generate a JSON source node that will read the exported data file back into the stream. For more information, see [JSON Source node](#).

Common Export node tabs

The following options can be specified for all export nodes by clicking the corresponding tab:

- Publish tab. Used to publish results of a stream.
- Annotations tab. Used for all nodes, this tab offers options to rename nodes, supply a custom ToolTip, and store a lengthy annotation.
- [**Publishing streams**](#)

Publishing streams

Publishing streams is done directly from IBM® SPSS® Modeler using any of the standard export nodes: Database, Flat File, Statistics Export, Extension Export, Data Collection Export, SAS Export, Excel, and XML Export nodes. The type of export node determines the format of the results to be written each time the published stream is executed using the IBM SPSS Modeler Solution Publisher Runtime or external application. For example, if you want to write your results to a database each time the published stream is run, use a Database export node.

To publish a stream

1. Open or build a stream in the normal manner and attach an export node at the end.
2. On the Publish tab in the export node, specify a root name for the published files (that is, the file name to which the .pim, .par, and .xml extensions will be appended).
3. Click Publish to publish the stream, or select Publish the stream to automatically publish the stream each time the node is executed.

Published name. Specify the root name for the published image and parameter files.

- The image file (*.pim) provides all of the information needed for the Runtime to execute the published stream exactly as it was at the time of export. If you are confident that you will not need to change any of the settings for the stream (such as the input data source or the output data file), you can deploy the image file only.
- The parameter file (*.par) contains configurable information about data sources, output files, and execution options. If you want to be able to control the input or output of the stream without republishing the stream, you will need the parameter file as well as the image file.
- The metadata file (*.xml) describes the inputs and outputs of the image and their data models. It is designed for use by applications which embed the runtime library and which need to know the structure of the input and output data.

Note: This file is only produced if you select the Publish metadata option.

Publish parameters. If required, you can include stream parameters in the *.par file. You can change these stream parameter values when you execute the image either by editing the *.par file or through the runtime API.

This option enables the Parameters button. The Publish Parameters dialog box is displayed when you click the button.

Choose the parameters you want to include in the published image by selecting the relevant option in the Publish column.

On stream execution. Specifies whether the stream is automatically published when the node is executed.

- Export data. Executes the export node in the standard manner, without publishing the stream. (Basically, the node executes in IBM SPSS Modeler the same way it would if IBM SPSS Modeler Solution Publisher were not available.) If you select this option, the stream will not be published unless you do so explicitly by clicking Publish in the export node dialog box. Alternatively, you can publish the current stream using the Publish tool on the toolbar or by using a script.
- Publish the stream. Publishes the stream for deployment using IBM SPSS Modeler Solution Publisher. Select this option if you want to automatically publish the stream every time it is executed.

Note:

- If you plan to run the published stream with new or updated data, it is important to note that the order of fields in the input file must be the same as the order of fields in the source node input file specified in the published stream.
- When publishing to external applications, consider filtering extraneous fields or renaming fields to conform with input requirements. Both can be accomplished using a Filter node prior to the export node.

IBM® SPSS® Statistics Nodes

- [IBM SPSS Statistics Nodes - Overview](#)
- [Statistics File Node](#)
- [Statistics Transform Node](#)
- [Statistics Model Node](#)
- [Statistics Output Node](#)
- [Statistics Export Node](#)

IBM SPSS Statistics Nodes - Overview

To complement IBM® SPSS® Modeler and its data mining abilities, IBM SPSS Statistics provides you with the ability to carry out further statistical analysis and data management.

If you have a compatible, licensed copy of IBM SPSS Statistics installed, you can connect to it from IBM SPSS Modeler and carry out complex, multistep data manipulation and analysis not otherwise supported by IBM SPSS Modeler. For the advanced user there is also the option to further modify the analysis by using command syntax. See the Release Notes for information on version compatibility.

If available, the IBM SPSS Statistics nodes are shown on a dedicated part of the nodes palette.

Note: We recommend that you instantiate your data in a Type node before using the IBM SPSS Statistics Transform, Model, or Output nodes. This is also a requirement when using the AUTORECODE syntax command.

The IBM SPSS Statistics palette contains the following nodes:

	The Statistics File node reads data from the .sav or .zsav file format used by IBM SPSS Statistics, as well as cache files saved in IBM SPSS Modeler, which also use the same format.
	The Statistics Transform node runs a selection of IBM SPSS Statistics syntax commands against data sources in IBM SPSS Modeler. This node requires a licensed copy of IBM SPSS Statistics.
	The Statistics Model node enables you to analyze and work with your data by running IBM SPSS Statistics procedures that produce PMML. This node requires a licensed copy of IBM SPSS Statistics.
	The Statistics Output node allows you to call an IBM SPSS Statistics procedure to analyze your IBM SPSS Modeler data. A wide variety of IBM SPSS Statistics analytical procedures is available. This node requires a licensed copy of IBM SPSS Statistics.
	The Statistics Export node outputs data in IBM SPSS Statistics .sav or .zsav format. The .sav or .zsav files can be read by IBM SPSS Statistics Base and other products. This is also the format used for cache files in IBM SPSS Modeler.

Note: If your copy of SPSS Statistics is licensed for a single user only and you run a stream with two or more branches, each of which contains an SPSS Statistics node, you may receive a licensing error. This occurs when the SPSS Statistics session for one branch has not terminated before the session for another branch attempts to start. If possible, redesign the stream so that multiple branches with SPSS Statistics nodes do not execute in parallel.

Related information

- [Statistics File Node](#)
- [Statistics Transform Node](#)
- [Statistics Output Node](#)
- [Statistics Model Node](#)
- [Statistics Export Node](#)

Statistics File Node

You can use the Statistics File node to read data directly from a saved IBM® SPSS® Statistics file (.sav or .zsav). This format is now used to replace the cache file from earlier versions of IBM SPSS Modeler. If you would like to import a saved cache file, you should use the IBM SPSS Statistics File node.

Import file. Specify the name of the file. You can enter a filename or click the ellipsis button (...) to select a file. The file path is shown once you have selected a file.

File is password encrypted. Select this box if you know the file is password protected; you are prompted to enter the Password. If the file is password protected, and you do not enter the password, a warning message is displayed if you attempt to change to another tab, refresh the data, preview the node contents, or try to execute a stream containing the node.

Note: Password protected files can only be opened by IBM SPSS Modeler version 16 or greater.

Variable names. Select a method of handling variable names and labels upon import from an IBM SPSS Statistics .sav or .zsav file. Metadata that you choose to include here persists throughout your work in IBM SPSS Modeler and may be exported again for use in IBM SPSS Statistics.

- **Read names and labels.** Select to read both variable names and labels into IBM SPSS Modeler. By default, this option is selected and variable names are displayed in the Type node. Labels may be displayed in charts, model browsers, and other types of output, depending on the options specified in the stream properties dialog box. By default, the display of labels in output is disabled.
- **Read labels as names.** Select to read the descriptive variable labels from the IBM SPSS Statistics .sav or .zsav file rather than the short field names, and use these labels as variable names in IBM SPSS Modeler.

Values. Select a method of handling values and labels upon import from an IBM SPSS Statistics .sav or .zsav file. Metadata that you choose to include here persists throughout your work in IBM SPSS Modeler and may be exported again for use in IBM SPSS Statistics.

- **Read data and labels.** Select to read both actual values and value labels into IBM SPSS Modeler. By default, this option is selected and values themselves are displayed in the Type node. Value labels may be displayed in the Expression Builder, charts, model browsers, and other types of output, depending on the options specified in the stream properties dialog box.
- **Read labels as data.** Select if you want to use the value labels from the .sav or .zsav file rather than the numerical or symbolic codes used to represent the values. For example, selecting this option for data with a gender field whose values of 1 and 2 actually represent *male* and *female*, respectively, will convert the field to a string and import *male* and *female* as the actual values.

It is important to consider missing values in your IBM SPSS Statistics data before selecting this option. For example, if a numeric field uses labels only for missing values (0 = *No Answer*, -99 = *Unknown*), then selecting the option above will import only the value labels *No Answer* and *Unknown* and will convert the field to a string. In such cases, you should import the values themselves and set missing values in a Type node.

Use field format information to determine storage. If this box is cleared, field values that are formatted in the .sav file as integers (i.e., fields specified as Fn.0 in the Variable View in IBM SPSS Statistics) are imported using integer storage. All other field values except strings are imported as real numbers.

If this box is selected (default), all field values except strings are imported as real numbers, whether formatted in the .sav file as integers or not.

Multiple response sets. Any multiple response sets defined in the IBM SPSS Statistics file will automatically be preserved when the file is imported. You can view and edit multiple response sets from any node with a Filter tab. See the topic [Editing Multiple Response Sets](#) for more information.

Statistics Transform Node

The Statistics Transform node enables you to complete data transformations using IBM® SPSS® Statistics command syntax. This makes it possible to complete a number of transformations not supported by IBM SPSS Modeler and allows automation of complex, multistep transformations, including the creation of a number of fields from a single node. It resembles the Statistics Output node, except that the data are returned to IBM SPSS Modeler for further analysis, whereas, in the Output node the data are returned as the requested output objects, such as graphs or tables.

You must have a compatible version of IBM SPSS Statistics installed and licensed on your computer to use this node. See [IBM SPSS Statistics Helper Applications](#) for more information. See the online Release Notes for compatibility information.

If necessary, you can use the Filter tab to filter or rename fields so they conform to IBM SPSS Statistics naming standards. See [Renaming or Filtering Fields for IBM SPSS Statistics](#) for more information.

Syntax Reference. For details about specific IBM SPSS Statistics procedures, see the *IBM SPSS Statistics Command Syntax Reference* guide, included with your copy of the IBM SPSS Statistics software. To view the guide from the Syntax tab, choose the Syntax editor option and click the Launch Statistics Syntax Help button.

Note: Not all IBM SPSS Statistics syntax is supported by this node. See the topic [Allowable Syntax](#) for more information.

- [Statistics Transform Node - Syntax Tab](#)
- [Allowable Syntax](#)

Related information

- [Statistics Transform Node - Syntax Tab](#)
- [Allowable Syntax](#)
- [Statistics File Node](#)
- [Statistics Output Node](#)
- [Statistics Model Node](#)

- [Statistics Export Node](#)

Statistics Transform Node - Syntax Tab

IBM® SPSS® Statistics dialog option

If you are unfamiliar with IBM SPSS Statistics syntax for a procedure, the simplest way to create syntax in IBM SPSS Modeler is to choose the IBM SPSS Statistics dialog option, select the dialog box for the procedure, complete the dialog box and click OK. Doing so places the syntax onto the Syntax tab of the IBM SPSS Statistics node you are using in IBM SPSS Modeler. You can then run the stream to obtain the output from the procedure.

IBM SPSS Statistics Syntax editor option

Check. After you have entered your syntax commands in the upper part of the dialog box, use this button to validate your entries. Any incorrect syntax is identified in the bottom part of the dialog box.

To ensure that the checking process does not take too long, when you validate the syntax, it is checked against a representative sample of your data to ensure that the entries are valid instead of checking against the entire dataset.

Related information

- [Statistics Transform Node](#)
- [Allowable Syntax](#)

Allowable Syntax

If you have a lot of legacy syntax from IBM® SPSS® Statistics or are familiar with the data preparation features of IBM SPSS Statistics, you can use the Statistics Transform node to run many of your existing transformations. As a guideline, the node enables you to transform data in predictable ways--for example, by running looped commands or by changing, adding, sorting, filtering, or selecting data.

Examples of the commands that can be carried out are:

- Compute random numbers according to a binomial distribution:

```
COMPUTE newvar = RV.BINOM(10000, 0.1)
```

- Recode a variable into a new variable:

```
RECODE Age (Lowest thru 30=1) (30 thru 50=2) (50 thru Highest=3) INTO AgeRecoded
```

- Replace missing values:

```
RMV Age_1=SMEAN(Age)
```

The IBM SPSS Statistics syntax that is supported by the Statistics Transform node is listed below.

Command Name
ADD VALUE LABELS
APPLY DICTIONARY
AUTORECODE
BREAK
CD
CLEAR MODEL PROGRAMS
CLEAR TIME PROGRAM
CLEAR TRANSFORMATIONS
COMPUTE
COUNT
CREATE
DATE
DEFINE-!ENDDEFINE
DELETE VARIABLES
DO IF
DO REPEAT
ELSE
ELSE IF
END CASE

Command Name
END FILE
END IF
END INPUT PROGRAM
END LOOP
END REPEAT
EXECUTE
FILE HANDLE
FILE LABEL
FILE TYPE-END FILE TYPE
FILTER
FORMATS
IF
INCLUDE
INPUT PROGRAM-END INPUT PROGRAM
INSERT
LEAVE
LOOP-END LOOP
MATRIX-END MATRIX
MISSING VALUES
N OF CASES
NUMERIC
PERMISSIONS
PRESERVE
RANK
RECODE
RENAME VARIABLES
RESTORE
RMV
SAMPLE
SELECT IF
SET
SORT CASES
STRING
SUBTITLE
TEMPORARY
TITLE
UPDATE
V2C
VALIDATEDATA
VALUE LABELS
VARIABLE ATTRIBUTE
VARSTOCASES
VECTOR

Related information

- [Statistics Transform Node](#)
 - [Statistics Transform Node - Syntax Tab](#)
-

Statistics Model Node

The Statistics Model node enables you to analyze and work with your data by running IBM® SPSS® Statistics procedures that produce PMML. The model nuggets you create can then be used in the usual way within IBM SPSS Modeler streams for scoring, and so on.

You must have a compatible version of IBM SPSS Statistics installed and licensed on your computer to use this node. See [IBM SPSS Statistics Helper Applications](#) for more information. See the online Release Notes for compatibility information.

The IBM SPSS Statistics analytical procedures that are available depend on the type of license you have.

- [Statistics Model Node - Model Tab](#)
- [Statistics Model Node - Model Nugget Summary](#)

Related information

- [Statistics Model Node - Model Tab](#)
 - [Statistics Model Node - Model Nugget Summary](#).
 - [Statistics File Node](#)
 - [Statistics Transform Node](#)
 - [Statistics Output Node](#)
 - [Statistics Export Node](#)
-

Statistics Model Node - Model Tab

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Select a dialog. Click to display a list of available IBM® SPSS® Statistics procedures that you can select and run. The list shows only those procedures that produce PMML and for which you are licensed, and does not include user-written procedures.

1. Click on the required procedure; the relevant IBM SPSS Statistics dialog is displayed.
2. In the IBM SPSS Statistics dialog, enter the details for the procedure.
3. Click OK to return to the Statistics Model node; the IBM SPSS Statistics syntax is displayed in the Model tab.
4. To return to the IBM SPSS Statistics dialog at any time, for example to modify your query, click the IBM SPSS Statistics dialog display button to the right of the procedure selection button.

Figure 1. IBM SPSS Statistics dialog display button



Related information

- [Statistics Model Node](#)
 - [Statistics Model Node - Model Nugget Summary](#).
-

Statistics Model Node - Model Nugget Summary

When you run the Statistics Model node, it executes the associated IBM® SPSS® Statistics procedure and creates a model nugget that you can use in IBM SPSS Modeler streams for scoring.

The Summary tab of the model nugget displays information about the fields, build settings, and model estimation process. Results are presented in a tree view that can be expanded or collapsed by clicking specific items.

The View Model button displays the results in a modified form of the IBM SPSS Statistics Output Viewer. For more information about this viewer, see the IBM SPSS Statistics documentation.

The usual exporting and printing options are available from the File menu. See the topic [Viewing output](#) for more information.

Related information

- [Statistics Model Node](#)
 - [Statistics Model Node - Model Tab](#)
-

Statistics Output Node

The Statistics Output node enables you to call an IBM® SPSS® Statistics procedure to analyze your IBM SPSS Modeler data. You can view the results in a browser window or save results in the IBM SPSS Statistics output file format. A wide variety of IBM SPSS Statistics analytical procedures is accessible from IBM SPSS Modeler.

You must have a compatible version of IBM SPSS Statistics installed and licensed on your computer to use this node. See [IBM SPSS Statistics Helper Applications](#) for more information. See the online Release Notes for compatibility information.

If necessary, you can use the Filter tab to filter or rename fields so they conform to IBM SPSS Statistics naming standards. See [Renaming or Filtering Fields for IBM SPSS Statistics](#) for more information.

Syntax Reference. For details about specific IBM SPSS Statistics procedures, see the *IBM SPSS Statistics Command Syntax Reference* guide, included with your copy of the IBM SPSS Statistics software. To view the guide from the Syntax tab, choose the Syntax editor option and click the

Launch Statistics Syntax Help button.

- [Statistics Output Node - Syntax Tab](#)
- [Statistics Output Node - Output Tab](#)

Related information

- [Overview of Output Nodes](#)
 - [Statistics Output Node - Syntax Tab](#)
 - [Statistics Output Node - Output Tab](#)
 - [Renaming or Filtering Fields for IBM SPSS Statistics](#)
 - [Statistics File Node](#)
 - [Statistics Transform Node](#)
 - [Statistics Model Node](#)
 - [Statistics Export Node](#)
-

Statistics Output Node - Syntax Tab

Use this tab to create syntax for the SPSS® Statistics procedure you want to use to analyze your data. Syntax is composed of two parts: a **statement** and associated **options**. The statement specifies the analysis or operation to be performed and the fields to be used. The options specify everything else, including which statistics to display, derived fields to save, and so on.

SPSS Statistics dialog option

If you are unfamiliar with IBM® SPSS Statistics syntax for a procedure, the simplest way to create syntax in IBM SPSS Modeler is to choose the IBM SPSS Statistics dialog option, select the dialog box for the procedure, complete the dialog box and click OK. Doing so places the syntax onto the Syntax tab of the IBM SPSS Statistics node you are using in IBM SPSS Modeler. You can then run the stream to obtain the output from the procedure.

You can optionally generate a Statistics File source node for importing the resulting data. This is useful, for example, if a procedure writes fields such as scores to the active dataset in addition to displaying output.

Note:

- When generating output in languages other than English, it is advisable to specify the language in the syntax.
- The Output Style option is not supported in the Statistics Output node.

To create the syntax

1. Click the Select a dialog button.
2. Choose one of the options:
 - Analyze Lists the contents of the SPSS Statistics Analyze menu; choose the procedure you want to use.
 - Other If shown, lists dialogs created by the Custom Dialog Builder in SPSS Statistics, as well as any other SPSS Statistics dialogs that do not appear on the Analyze menu and for which you have a licence. If there are no applicable dialogs, this option is not shown.

Note: The Automatic Data Preparation dialogs are not shown.

If you have an SPSS Statistics custom dialog that creates new fields, these fields cannot be used in SPSS Modeler because the Statistics Output node is a terminal node.

Optionally check the Generate an import node for the resulting data box to create a Statistics File source node that can be used to import the resulting data into another stream. The node is placed on the screen canvas, with the data contained in the .sav file specified in the File field (default location is the SPSS Modeler installation directory).

Syntax editor option

To save syntax that has been created for a frequently used procedure:

1. Click the File Options button (the first one on the toolbar).
2. Choose Save or Save As from the menu.
3. Save the file as an .sps file.

To use previously created syntax files, replacing the current contents, if any, of the Syntax editor:

1. Click the File Options button (the first one on the toolbar).

2. Choose Open from the menu.
3. Select an .sps file to paste its contents into the Output node Syntax tab.

To insert previously saved syntax without replacing the current contents:

1. Click the File Options button (the first one on the toolbar).
2. Choose Insert from the menu
3. Select an .sps file to paste its contents into the Output node at the point specified by the cursor.

Optionally check the Generate an import node for the resulting data box to create a Statistics File source node that can be used to import the resulting data into another stream. The node is placed on the screen canvas, with the data contained in the .sav file specified in the File field (default location is the SPSS Modeler installation directory).

When you click Run, the results are shown in the SPSS Statistics Output Viewer. For more information about the viewer, see the SPSS Statistics documentation.

Note: Syntax for the following items (and their corresponding SPSS Statistics dialog box options) is not supported. They have no impact on the output.

- **OUTPUT ACTIVATE**
- **OUTPUT CLOSE**
- **OUTPUT DISPLAY**
- **OUTPUT EXPORT**
- **OUTPUT MODIFY**
- **OUTPUT NAME**
- **OUTPUT NEW**
- **OUTPUT OPEN**
- **OUTPUT SAVE**

Statistics Output Node - Output Tab

The Output tab lets you specify the format and location of the output. You can choose to display the results on the screen or send them to one of the available file types.

Output name. Specifies the name of the output produced when the node is executed. Auto chooses a name based on the node that generates the output. Optionally, you can select Custom to specify a different name.

Output to screen (the default). Creates an output object to view online. The output object will appear on the Outputs tab of the manager window when the output node is executed.

Output to file. Saves the output to a file when you run the node. If you choose this option, enter a filename in the Filename field (or navigate to a directory and specify a filename using the File Chooser button) and select a file type.

File type. Choose the type of file to which you want to send the output.

- **HTML document (*.html).** Writes the output in HTML format.
- **IBM® SPSS® Statistics Viewer File (*.spv).** Writes the output in a format that can be read by the IBM SPSS Statistics Output Viewer.
- **IBM SPSS Statistics Web Reports File (*.spw).** Writes the output in IBM SPSS Statistics Web Reports format, which can be published to an IBM SPSS Collaboration and Deployment Services repository and subsequently viewed in a Web browser. See the topic [Publish to Web](#) for more information.

Note: If you select Output to screen the IBM SPSS Statistics OMS directive **VIEWER=NO** has no effect; also, the scripting APIs (*Basic* and *Python SpssClient* module) are not available in IBM SPSS Modeler.

Related information

- [Viewing output](#)
- [Statistics Output Node](#)
- [Statistics Output Node - Syntax Tab](#)
- [Renaming or Filtering Fields for IBM SPSS Statistics](#)

Statistics Export Node

The Statistics Export node enables you to export data in IBM® SPSS® Statistics .sav format. IBM SPSS Statistics .sav files can be read by IBM SPSS Statistics Base and other modules. This is also the format used for IBM SPSS Modeler cache files.

Mapping IBM SPSS Modeler field names to IBM SPSS Statistics variable names can sometimes cause errors because IBM SPSS Statistics variable names are limited to 64 characters and cannot include certain characters, such as spaces, dollar signs (\$), and dashes (-). There are two ways to adjust for these restrictions:

- You can rename fields conforming to IBM SPSS Statistics variable name requirements by clicking the Filter tab. See the topic [Renaming or Filtering Fields for IBM SPSS Statistics](#) for more information.
- Choose to export both field names and labels from IBM SPSS Modeler.

Note: IBM SPSS Modeler writes .sav files in Unicode UTF-8 format. IBM SPSS Statistics only supports files in Unicode UTF-8 format from release 16.0 onwards. To prevent the possibility of data corruption .sav files saved with Unicode encoding should not be used in releases of IBM SPSS Statistics prior to 16.0. For more information, see the IBM SPSS Statistics help.

Multiple response sets. Any multiple response sets defined in the stream will automatically be preserved when the file is exported. You can view and edit multiple response sets from any node with a Filter tab. See the topic [Editing Multiple Response Sets](#) for more information.

- [Statistics Export Node - Export Tab](#)
- [Renaming or Filtering Fields for IBM SPSS Statistics](#)
- [Statistics Export Node - Export Tab](#)
- [Renaming or Filtering Fields for IBM SPSS Statistics](#)

Related information

- [Statistics Export Node - Export Tab](#)
- [Statistics File Node](#)
- [Statistics Transform Node](#)
- [Statistics Output Node](#)
- [Statistics Model Node](#)
- [Overview of Output Nodes](#)

Statistics Export Node - Export Tab

Export file Specifies the name of the file. Enter a filename or click the file chooser button to browse to the file's location.

File type Select if the file is to be saved in normal .sav or compressed.zsav format.

Encrypt file with password To protect the file with a password, select this box; you are prompted to enter and confirm the Password in a separate dialog box.

Note: Password protected files can only be opened by SPSS® Modeler version 16 or greater, or by SPSS Statistics version 21 or greater.

Export field names Specifies a method of handling variable names and labels upon export from SPSS Modeler to an SPSS Statistics .sav or .zsav file.

- Names and variable labels Select to export both SPSS Modeler field names and field labels. Names are exported as SPSS Statistics variable names, while labels are exported as SPSS Statistics variable labels.
- Names as variable labels Select to use the SPSS Modeler field names as variable labels in SPSS Statistics. SPSS Modeler allows characters in field names that are invalid in SPSS Statistics variable names. To prevent possibly creating invalid SPSS Statistics names, select Names as variable labels instead, or use the Filter tab to adjust field names.

Launch Application If SPSS Statistics is installed on your computer, you can select this option to invoke the application directly on the saved data file. Options for launching the application must be specified in the Helper Applications dialog box. See the topic [IBM SPSS Statistics Helper Applications](#) for more information. To simply create an SPSS Statistics .sav or .zsav file without opening an external program, deselect this option.

Note: When running SPSS Modeler and SPSS Statistics together in Server (distributed) mode, writing the data out and launching a SPSS Statistics session does not automatically open a SPSS Statistics client showing the data set read into the active data set. The workaround is to manually open the data file in the SPSS Statistics client once it is launched.

Generate an import node for this data Select this option to automatically generate a Statistics File source node that will read the exported data file. See the topic [Statistics File Node](#) for more information.

Related information

- [Statistics Export Node](#)

Renaming or Filtering Fields for IBM SPSS Statistics

Before exporting or deploying data from IBM® SPSS® Modeler to external applications such as IBM SPSS Statistics, it may be necessary to rename or adjust field names. The Statistics Transform, Statistics Output, and Statistics Export dialog boxes contain a Filter tab to facilitate this process.

A basic description of Filter tab functionality is discussed elsewhere. See the topic [Setting filtering options](#) for more information.

To adjust field names to conform to IBM SPSS Statistics naming conventions:

1. On the Filter tab, click the Filter Options Menu toolbar button (the first one on the toolbar).
2. Select Rename For IBM SPSS Statistics.
3. On the Rename For IBM SPSS Statistics dialog, you can choose to replace invalid characters in filenames with either a Hash (#) character or an Underscore (_).

Rename multi response sets. Select this option if you want to adjust the names of multiple response sets, which can be imported into IBM SPSS Modeler using a Statistics File source node. They are used to record data that can have more than one value for each case, such as in survey responses.

Related information

- [Statistics Export Node](#)
 - [Statistics Output Node](#)
 - [Statistics Output Node - Syntax Tab](#)
 - [Statistics Output Node - Output Tab](#)
-

SuperNodes

- [Overview of SuperNodes](#)
 - [Types of SuperNodes](#)
 - [Creating SuperNodes](#)
 - [Locking SuperNodes](#)
 - [Editing SuperNodes](#)
 - [Saving and loading SuperNodes](#)
-

Overview of SuperNodes

One of the reasons that the IBM® SPSS® Modeler visual programming interface is so easy to learn is that each node has a clearly defined function. However, for complex processing, a long sequence of nodes may be necessary. Eventually, this may clutter the stream canvas and make it difficult to follow stream diagrams. There are two ways to avoid the clutter of a long and complex stream:

- You can split a processing sequence into several streams that feed one into the other. The first stream, for example, creates a data file that the second uses as input. The second creates a file that the third uses as input, and so on. You can manage these multiple streams by saving them in a **project**. A project provides organization for multiple streams and their output. However, a project file contains only a reference to the objects it contains, and you will still have multiple stream files to manage.
- You can create a **SuperNode** as a more streamlined alternative when working with complex stream processes.

SuperNodes group multiple nodes into a single node by encapsulating sections of a data stream. This provides numerous benefits to the data miner:

- Streams are neater and more manageable.
 - Nodes can be combined into a business-specific SuperNode.
 - SuperNodes can be exported to libraries for reuse in multiple data mining projects.
-

Types of SuperNodes

SuperNodes are represented in the data stream by a star icon. The icon is shaded to represent the type of SuperNode and the direction in which the stream must flow to or from it.

There are three types of SuperNodes:

- Source SuperNodes
- Process SuperNodes
- Terminal SuperNodes

- [Source SuperNodes](#)
 - [Process SuperNodes](#)
 - [Terminal SuperNodes](#)
-

Source SuperNodes

Source SuperNodes contain a data source just like a normal source node and can be used anywhere that a normal source node can be used. The left side of a source SuperNode is shaded to indicate that it is "closed" on the left and that data must flow downstream *from* a SuperNode.

Source SuperNodes have only one connection point on the right, showing that data leaves the SuperNode and flows to the stream.

Related information

- [Creating SuperNodes](#)
 - [Types of SuperNodes](#)
-

Process SuperNodes

Process SuperNodes contain only process nodes and are unshaded to show that data can flow both *in* and *out* of this type of SuperNode.

Process SuperNodes have connection points on both the left and right, showing that data enters the SuperNode and leaves to flow back to the stream. Although SuperNodes can contain additional stream fragments and even extra streams, both connection points must flow through a single path connecting the *From Stream* and *To Stream* points.

Note: Process SuperNodes are also sometimes referred to as *Manipulation SuperNodes*.

Related information

- [Creating SuperNodes](#)
 - [Types of SuperNodes](#)
-

Terminal SuperNodes

Terminal SuperNodes contain one or more terminal nodes (plot, table, and so on) and can be used in the same manner as a terminal node. A terminal SuperNode is shaded on the right side to indicate that it is "closed" on the right and that data can flow only *into* a terminal SuperNode.

Terminal SuperNodes have only one connection point on the left, showing that data enters the SuperNode from the stream and terminates inside the SuperNode.

Terminal SuperNodes can also contain scripts that are used to specify the order of execution for all terminal nodes inside the SuperNode. See the topic [SuperNodes and scripting](#) for more information.

Related information

- [Creating SuperNodes](#)
 - [Types of SuperNodes](#)
-

Creating SuperNodes

Creating a SuperNode "shrinks" the data stream by encapsulating several nodes into one node. Once you have created or loaded a stream on the canvas, there are several ways to create a SuperNode.

Multiple Selection

The simplest way to create a SuperNode is by selecting all of the nodes that you want to encapsulate:

1. Use the mouse to select multiple nodes on the stream canvas. You can also use Shift-click to select a stream or section of a stream.
Note: Nodes that you select must be from a continuous or forked stream. You cannot select nodes that are not adjacent or connected in some way.
2. Then, using one of the following three methods, encapsulate the selected nodes:
 - Click the SuperNode icon (shaped like a star) on the toolbar.
 - Right-click the SuperNode, and from the context menu choose:
Create SuperNode > From Selection
 - From the SuperNode menu, choose:
Create SuperNode > From Selection

All three of these options encapsulate the nodes into a SuperNode shaded to reflect its type--source, process, or terminal--based on its contents.

Single Selection

You can also create a SuperNode by selecting a single node and using menu options to determine the start and end of the SuperNode or encapsulating everything downstream of the selected node.

1. Click the node that determines the start of encapsulation.
2. From the SuperNode menu, choose:
Create SuperNode > From Here

SuperNodes can also be created more interactively by selecting the start and end of the stream section to encapsulate nodes:

1. Click on the first or last node that you want to include in the SuperNode.
2. From the SuperNode menu, choose:
Create SuperNode > Select...
3. Alternatively, you can use the context menu options by right-clicking the desired node.
4. The cursor becomes a SuperNode icon, indicating that you must select another point in the stream. Move either upstream or downstream to the "other end" of the SuperNode fragment and click on a node. This action will replace all nodes in between with the SuperNode star icon.

Note: Nodes that you select must be from a continuous or forked stream. You cannot select nodes that are not adjacent or connected in some way.

- [Nesting SuperNodes](#)
-

Nesting SuperNodes

SuperNodes can be nested within other SuperNodes. The same rules for each type of SuperNode (source, process, and terminal) apply to nested SuperNodes. For example, a process SuperNode with nesting must have a continuous data flow through all nested SuperNodes in order for it to remain a process SuperNode. If one of the nested SuperNodes is terminal, then data would no longer flow through the hierarchy.

Terminal and source SuperNodes can contain other types of nested SuperNodes, but the same basic rules for creating SuperNodes apply.

Related information

- [Creating SuperNodes](#)
-

Locking SuperNodes

Once you have created a SuperNode, you can lock it with a password to prevent it from being amended. For example, you might do this if you are creating streams, or parts of streams, as fixed-value templates for use by others in your organization who have less experience with setting up IBM® SPSS® Modeler enquiries.

When a SuperNode is locked users can still enter values on the Parameters tab for any parameters that have been defined, and a locked SuperNode can be executed without entering the password.

Note: Locking and unlocking cannot be performed using scripts.

- [Locking and unlocking a SuperNode](#)
- [Editing a locked SuperNode](#)

Locking and unlocking a SuperNode

Warning: Lost passwords cannot be recovered.

You can lock or unlock a SuperNode from any of the three tabs.

1. Click Lock Node.
2. Enter and confirm the password.
3. Click OK.

A password protected SuperNode is identified on the stream canvas by a small padlock symbol to the top-left of the SuperNode icon.

Unlocking a SuperNode

1. To permanently remove the password protection, click Unlock Node. You are prompted for the password.
2. Enter the password and click OK. The SuperNode is no longer password protected and the padlock symbol no longer shows next to the icon in the stream.

For a stream saved in a version of SPSS® Modeler between 16 and 17.0 that contains a locked SuperNode, when opening the stream in a different environment such as in IBM® SPSS Collaboration and Deployment Services or on a Mac when the SPSS Modeler-installed JRE is different, it must first be opened, unlocked, and resaved using version 17.1 or later on the old environment where it was last saved.

Sometimes an incorrect password error will be displayed when unlocking a SuperNode in stream older than version 18. To work around this, reopen and unlock the node using the exact IBM SPSS Modeler version (or a more current version) on the same platform with the same system local settings as when it was last saved. Then open it in version 18 or later, lock the node, and save the stream again.

Editing a locked SuperNode

If you attempt to either define parameters or zoom in to display a locked SuperNode you are prompted to enter the password.

Enter the password and click OK.

You are now able to edit the parameter definitions and zoom in and out as often as you require until you close the stream that the SuperNode is in.

Note that this does not remove the password protection, it only allows you access to work with the SuperNode. See the topic [Locking and unlocking a SuperNode](#) for more information.

Editing SuperNodes

Once you have created a SuperNode, you can examine it more closely by zooming in to it; if the SuperNode is locked, you will be prompted to enter the password. See the topic [Editing a locked SuperNode](#) for more information.

To view the contents of a SuperNode, you can use the zoom-in icon from the IBM® SPSS® Modeler toolbar, or the following method:

1. Right-click a SuperNode.
2. From the context menu, choose Zoom In.

The contents of the selected SuperNode will be displayed in a slightly different IBM SPSS Modeler environment, with connectors showing the flow of data through the stream or stream fragment. At this level on the stream canvas, there are several tasks that you can perform:

- Modify the SuperNode type—source, process, or terminal.
 - Create parameters or edit the values of a parameter. Parameters are used in scripting and CLEM expressions.
 - Specify caching options for the SuperNode and its subnodes.
 - Create or modify a SuperNode script (terminal SuperNodes only).
- [Modifying SuperNode types](#)
 - [Annotating and renaming SuperNodes](#)
 - [SuperNode parameters](#)
 - [SuperNodes and caching](#)
 - [SuperNodes and scripting](#)

Modifying SuperNode types

In some circumstances, it is useful to alter the type of a SuperNode. This option is available only when you are zoomed in to a SuperNode, and it applies only to the SuperNode at that level. The three types of SuperNodes are explained in the following table.

Table 1. Types of SuperNode

Type of SuperNode	Description
Source SuperNode	One connection going out
Process SuperNode	Two connections: one coming in and one going out
Terminal SuperNode	One connection coming in

To change the type of a SuperNode

1. Be sure that you are zoomed in to the SuperNode.
2. From the SuperNode menu, choose SuperNode Type, and then choose the type.

Annotating and renaming SuperNodes

You can rename a SuperNode as it appears in the stream, and write annotations used in a project or report. To access these properties:

- Right-click a SuperNode (zoomed out) and choose Rename and Annotate.
- Alternatively, from the SuperNode menu, choose Rename and Annotate. This option is available in both zoomed-in and zoomed-out modes.

In both cases, a dialog box opens with the Annotations tab selected. Use the options here to customize the name displayed on the stream canvas and provide documentation regarding SuperNode operations.

Using Comments with SuperNodes

If you create a SuperNode from a commented node or nugget, you must include the comment in the selection to create the SuperNode if you want the comment to appear in the SuperNode. If you omit the comment from the selection, the comment will remain on the stream when the SuperNode is created.

When you expand a SuperNode that included comments, the comments are reinstated to where they were before the SuperNode was created.

When you expand a SuperNode that included commented objects, but the comments were not included in the SuperNode, the objects are reinstated to where they were, but the comments are not reattached.

SuperNode parameters

In IBM® SPSS® Modeler, you have the ability to set user-defined variables, such as **Minvalue**, whose values can be specified when used in scripting or CLEM expressions. These variables are called **parameters**. You can set parameters for streams, sessions, and SuperNodes. Any parameters set for a SuperNode are available when building CLEM expressions in that SuperNode or any nested nodes. Parameters set for nested SuperNodes are not available to their parent SuperNode.

There are two steps to creating and setting parameters for SuperNodes:

1. Define parameters for the SuperNode.
2. Then, specify the value for each parameter of the SuperNode.

These parameters can then be used in CLEM expressions for any encapsulated nodes.

- [Defining SuperNode Parameters](#)
- [Setting Values for SuperNode Parameters](#)
- [Using SuperNode Parameters to Access Node Properties](#)

Defining SuperNode Parameters

Parameters for a SuperNode can be defined in both zoomed-out and zoomed-in modes. The parameters defined apply to all encapsulated nodes. To define the parameters of a SuperNode, you first need to access the Parameters tab of the SuperNode dialog box. Use one of the following methods to open the dialog box:

- Double-click a SuperNode in the stream.
- From the SuperNode menu, choose Set Parameters.
- Alternatively, when zoomed in to a SuperNode, choose Set Parameters from the context menu.

Once you have opened the dialog box, the Parameters tab is visible with any previously defined parameters.

To Define a New Parameter

Click the Define Parameters button to open the dialog box.

Name. Parameter names are listed here. You can create a new parameter by entering a name in this field. For example, to create a parameter for the minimum temperature, you could type `minvalue`. Do not include the `$P-` prefix that denotes a parameter in CLEM expressions. This name is also used for display in the CLEM Expression Builder.

Long name. Lists the descriptive name for each parameter created.

Storage. Select a storage type from the list. Storage indicates how the data values are stored in the parameter. For example, when working with values containing leading zeros that you want to preserve (such as 008), you should select String as the storage type. Otherwise, the zeros will be stripped from the value. Available storage types are string, integer, real, time, date, and timestamp. For date parameters, note that values must be specified using ISO standard notation as shown in the next paragraph.

Value. Lists the current value for each parameter. Adjust the parameter as required. Note that for date parameters, values must be specified in ISO standard notation (that is, `YYYY-MM-DD`). Dates specified in other formats are not accepted.

Type (optional). If you plan to deploy the stream to an external application, select a measurement level from the list. Otherwise, it is advisable to leave the *Type* column as is. If you want to specify value constraints for the parameter, such as upper and lower bounds for a numeric range, select Specify from the list.

Note that long name, storage, and type options can be set for parameters through the user interface only. These options cannot be set using scripts.

Click the arrows at the right to move the selected parameter further up or down the list of available parameters. Use the delete button (marked with an *X*) to remove the selected parameter.

Related information

- [SuperNode parameters](#)
 - [Setting Values for SuperNode Parameters](#)
 - [Overview of SuperNodes](#)
 - [Using SuperNode Parameters to Access Node Properties](#)
-

Setting Values for SuperNode Parameters

Once you have defined parameters for a SuperNode, you can specify values using the parameters in a CLEM expression or script.

To Specify the Parameters of a SuperNode

1. Double-click the SuperNode icon to open the SuperNode dialog box.
2. Alternatively, from the SuperNode menu, choose Set Parameters.
3. Click the Parameters tab. *Note:* The fields in this dialog box are the fields defined by clicking the Define Parameters button on this tab.
4. Enter a value in the text box for each parameter that you have created. For example, you can set the value `minvalue` to a particular threshold of interest. This parameter can then be used in numerous operations, such as selecting records above or below this threshold for further exploration.

Related information

- [Defining SuperNode Parameters](#)
 - [SuperNode parameters](#)
 - [Overview of SuperNodes](#)
 - [Using SuperNode Parameters to Access Node Properties](#)
-

Using SuperNode Parameters to Access Node Properties

SuperNode parameters can also be used to define node properties (also known as **slot parameters**) for encapsulated nodes. For example, suppose you want to specify that a SuperNode train an encapsulated Neural Net node for a certain length of time using a random sample of the data available. Using parameters, you can specify values for the length of time and percentage sample.

Suppose your example SuperNode contains a Sample node called *Sample* and a Neural Net node called *Train*. You can use the node dialog boxes to specify the Sample node's **Sample** setting as Random % and the Neural Net node's **Stop on** setting to Time. Once these options are specified, you can access the node properties with parameters and specify specific values for the SuperNode. In the SuperNode dialog box, click Define Parameters and create the parameters shown in the following table.

Table 1. Parameters to create

Parameter	Value	Long name
Train.time	5	Time to train (minutes)
Sample.random	10	Percentage random sample

Note: The parameter names, such as *Sample.random*, use correct syntax for referring to node properties, where *Sample* represents the name of the node and *random* is a node property.

Once you have defined these parameters, you can easily modify values for the two Sample and Neural Net node properties without reopening each dialog box. Instead, simply select Set Parameters from the SuperNode menu to access the Parameters tab of the SuperNode dialog box, where you can specify new values for Random % and Time. This is particularly useful when exploring the data during numerous iterations of model building.

SuperNodes and caching

From within a SuperNode, all nodes except terminal nodes can be cached. Caching is controlled by right-clicking a node and choosing one of several options from the Cache context menu. This menu option is available both from outside a SuperNode and for the nodes encapsulated within a SuperNode.

There are several guidelines for SuperNode caches:

- If any of the nodes encapsulated in a SuperNode have caching enabled, the SuperNode will also.
- Disabling the cache on a SuperNode disables the cache for *all* encapsulated nodes.
- Enabling caching on a SuperNode actually enables the cache on the last cacheable subnode. In other words, if the last subnode is a Select node, the cache will be enabled for that Select node. If the last subnode is a terminal node (which does not allow caching), the next node upstream that supports caching will be enabled.
- Once you have set caches for the subnodes of a SuperNode, any activities upstream from the cached node, such as adding or editing nodes, will flush the caches.

SuperNodes and scripting

You can use the SPSS® Modeler scripting language to write simple programs that manipulate and execute the contents of a terminal SuperNode. For instance, you might want to specify the order of execution for a complex stream. As an example, if a SuperNode contains a Set Globals node that needs to be executed before a Plot node, you can create a script that executes the Set Globals node first. Values calculated by this node, such as the average or standard deviation, can then be used when the Plot node is executed.

The Script tab of the SuperNode dialog box is available only for terminal SuperNodes.

To open the Scripting dialog box for a Terminal SuperNode:

- Right-click the SuperNode canvas and choose SuperNode Script.
- Alternatively, in both zoomed-in and zoomed-out modes, you can choose SuperNode Script from the SuperNode menu.

Note: SuperNode scripts are executed only with the stream and SuperNode when you have selected Run this script in the dialog box. Specific options for scripting and its use within SPSS Modeler are discussed in the *Scripting and Automation Guide*, which is available as a PDF file as part of your product download.

Saving and loading SuperNodes

One of the advantages of SuperNodes is that they can be saved and reused in other streams. When saving and loading SuperNodes, note that they use an .slb extension.

To save a SuperNode

1. Zoom in on the SuperNode.
2. From the SuperNode menu, choose Save SuperNode.
3. Specify a filename and directory in the dialog box.
4. Select whether to add the saved SuperNode to the current project.
5. Click Save.

To load a SuperNode

1. From the Insert menu in the IBM® SPSS® Modeler window, choose SuperNode.
2. Select a SuperNode file (.slb) from the current directory or browse to a different one.
3. Click Load.

Note: Imported SuperNodes have the default values for all of their parameters. To change the parameters, double-click a SuperNode on the stream canvas.

Modeling Overview

- [Overview of modeling nodes](#)
 - [Building Split Models](#)
 - [Modeling Node Fields Options](#)
 - [Modeling Node Analyze Options](#)
 - [Misclassification Costs](#)
 - [Model Nuggets](#)
 - [Generated Statistical Models Advanced Output](#)
 - [Cluster Models Model Tab](#)
 - [Generated Rule Set/Decision Tree Model Tab](#)
-

Overview of modeling nodes

IBM® SPSS® Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics. The methods available on the Modeling palette allow you to derive new information from your data and to develop predictive models. Each method has certain strengths and is best suited for particular types of problems.

The *IBM SPSS Modeler Applications Guide* provides examples for many of these methods, along with a general introduction to the modeling process. This guide is available as an online tutorial, and also in PDF format. [More information](#).

Modeling methods are divided into these categories:

- Supervised
- Association
- Segmentation

Supervised Models

Supervised models use the values of one or more **input** fields to predict the value of one or more output, or **target**, fields. Some examples of these techniques are: decision trees (C&R Tree, QUEST, CHAID and C5.0 algorithms), regression (linear, logistic, generalized linear, and Cox regression algorithms), neural networks, support vector machines, and Bayesian networks.

Supervised models help organizations to predict a known result, such as whether a customer will buy or leave or whether a transaction fits a known pattern of fraud. Modeling techniques include machine learning, rule induction, subgroup identification, statistical methods, and multiple model generation.

Supervised nodes

	The Auto Classifier node creates and compares a number of different models for binary outcomes (yes or no, churn or do not churn, and so on), allowing you to choose the best approach for a given analysis. A number of modeling algorithms are supported, making it possible to select the methods you want to use, the specific options for each, and the criteria for comparing the results. The node generates a set of models based on the specified options and ranks the best candidates according to the criteria you specify.
	The Auto Numeric node estimates and compares models for continuous numeric range outcomes using a number of different methods. The node works in the same manner as the Auto Classifier node, allowing you to choose the algorithms to use and to experiment with multiple combinations of options in a single modeling pass. Supported algorithms include neural networks, C&R Tree, CHAID, linear regression, generalized linear regression, and support vector machines (SVM). Models can be compared based on correlation, relative error, or number of variables used.

	The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered “pure” if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).
	The QUEST node provides a binary classification method for building decision trees, designed to reduce the processing time required for large C&R Tree analyses while also reducing the tendency found in classification tree methods to favor inputs that allow more splits. Input fields can be numeric ranges (continuous), but the target field must be categorical. All splits are binary.
	The CHAID node generates decision trees using chi-square statistics to identify optimal splits. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.
	The C5.0 node builds either a decision tree or a rule set. The model works by splitting the sample based on the field that provides the maximum information gain at each level. The target field must be categorical. Multiple splits into more than two subgroups are allowed.
	The Decision List node identifies subgroups, or segments, that show a higher or lower likelihood of a given binary outcome relative to the overall population. For example, you might look for customers who are unlikely to churn or are most likely to respond favorably to a campaign. You can incorporate your business knowledge into the model by adding your own custom segments and previewing alternative models side by side to compare the results. Decision List models consist of a list of rules in which each rule has a condition and an outcome. Rules are applied in order, and the first rule that matches determines the outcome.
	Linear regression models predict a continuous target based on linear relationships between the target and one or more predictors.
	The PCA/Factor node provides powerful data-reduction techniques to reduce the complexity of your data. Principal components analysis (PCA) finds linear combinations of the input fields that do the best job of capturing the variance in the entire set of fields, where the components are orthogonal (perpendicular) to each other. Factor analysis attempts to identify underlying factors that explain the pattern of correlations within a set of observed fields. For both approaches, the goal is to find a small number of derived fields that effectively summarizes the information in the original set of fields.
	The Feature Selection node screens input fields for removal based on a set of criteria (such as the percentage of missing values); it then ranks the importance of remaining inputs relative to a specified target. For example, given a data set with hundreds of potential inputs, which are most likely to be useful in modeling patient outcomes?
	Discriminant analysis makes more stringent assumptions than logistic regression but can be a valuable alternative or supplement to a logistic regression analysis when those assumptions are met.
	Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric range.
	The Generalized Linear model expands the general linear model so that the dependent variable is linearly related to the factors and covariates through a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers the functionality of a wide number of statistical models, including linear regression, logistic regression, loglinear models for count data, and interval-censored survival models.
	A generalized linear mixed model (GLMM) extends the linear model so that the target can have a non-normal distribution, is linearly related to the factors and covariates via a specified link function, and so that the observations can be correlated. Generalized linear mixed models cover a wide variety of models, from simple linear regression to complex multilevel models for non-normal longitudinal data.
	The Cox regression node enables you to build a survival model for time-to-event data in the presence of censored records. The model produces a survival function that predicts the probability that the event of interest has occurred at a given time (t) for given values of the input variables.
	The Support Vector Machine (SVM) node enables you to classify data into one of two groups without overfitting. SVM works well with wide data sets, such as those with a very large number of input fields.
	The Bayesian Network node enables you to build a probability model by combining observed and recorded evidence with real-world knowledge to establish the likelihood of occurrences. The node focuses on Tree Augmented Naïve Bayes (TAN) and Markov Blanket networks that are primarily used for classification.
	The Self-Learning Response Model (SLRM) node enables you to build a model in which a single new case, or small number of new cases, can be used to reestimate the model without having to retrain the model using all data.
	The Time Series node estimates exponential smoothing, univariate Autoregressive Integrated Moving Average (ARIMA), and multivariate ARIMA (or transfer function) models for time series data and produces forecasts of future performance. This Time Series node is similar to the previous Time Series node that was deprecated in SPSS Modeler version 18. However, this newer Time Series node is designed to harness the power of IBM SPSS Analytic Server to process big data, and display the resulting model in the output viewer that was added in SPSS Modeler version 17.
	The k -Nearest Neighbor (KNN) node associates a new case with the category or value of the k objects nearest to it in the predictor space, where k is an integer. Similar cases are near each other and dissimilar cases are distant from each other.



The Spatio-Temporal Prediction (STP) node uses data that contains location data, input fields for prediction (predictors), a time field, and a target field. Each location has numerous rows in the data that represent the values of each predictor at each time of measurement. After the data is analyzed, it can be used to predict target values at any location within the shape data that is used in the analysis.

Association Models

Association models find patterns in your data where one or more entities (such as events, purchases, or attributes) are associated with one or more other entities. The models construct rule sets that define these relationships. Here the fields within the data can act as both inputs and targets. You could find these associations manually, but association rule algorithms do so much more quickly, and can explore more complex patterns. Apriori and Carma models are examples of the use of such algorithms. One other type of association model is a sequence detection model, which finds sequential patterns in time-structured data.

Association models are most useful when predicting multiple outcomes—for example, customers who bought product X also bought Y and Z. Association models associate a particular conclusion (such as the decision to buy something) with a set of conditions. The advantage of association rule algorithms over the more standard decision tree algorithms (C5.0 and C&RT) is that associations can exist between any of the attributes. A decision tree algorithm will build rules with only a single conclusion, whereas association algorithms attempt to find many rules, each of which may have a different conclusion.

Association nodes

	The Apriori node extracts a set of rules from the data, pulling out the rules with the highest information content. Apriori offers five different methods of selecting rules and uses a sophisticated indexing scheme to process large data sets efficiently. For large problems, Apriori is generally faster to train; it has no arbitrary limit on the number of rules that can be retained, and it can handle rules with up to 32 preconditions. Apriori requires that input and output fields all be categorical but delivers better performance because it is optimized for this type of data.
	The CARMA model extracts a set of rules from the data without requiring you to specify input or target fields. In contrast to Apriori the CARMA node offers build settings for rule support (support for both antecedent and consequent) rather than just antecedent support. This means that the rules generated can be used for a wider variety of applications—for example, to find a list of products or services (antecedents) whose consequent is the item that you want to promote this holiday season.
	The Sequence node discovers association rules in sequential or time-oriented data. A sequence is a list of item sets that tends to occur in a predictable order. For example, a customer who purchases a razor and aftershave lotion may purchase shaving cream the next time he shops. The Sequence node is based on the CARMA association rules algorithm, which uses an efficient two-pass method for finding sequences.
	The Association Rules Node is similar to the Apriori Node; however, unlike Apriori, the Association Rules Node can process list data. In addition, the Association Rules Node can be used with IBM SPSS Analytic Server to process big data and take advantage of faster parallel processing.

Segmentation Models

Segmentation models divide the data into segments, or clusters, of records that have similar patterns of input fields. As they are only interested in the input fields, segmentation models have no concept of output or target fields. Examples of segmentation models are Kohonen networks, K-Means clustering, two-step clustering and anomaly detection.

Segmentation models (also known as "clustering models") are useful in cases where the specific result is unknown (for example, when identifying new patterns of fraud, or when identifying groups of interest in your customer base). Clustering models focus on identifying groups of similar records and labeling the records according to the group to which they belong. This is done without the benefit of prior knowledge about the groups and their characteristics, and it distinguishes clustering models from the other modeling techniques in that there is no predefined output or target field for the model to predict. There are no right or wrong answers for these models. Their value is determined by their ability to capture interesting groupings in the data and provide useful descriptions of those groupings. Clustering models are often used to create clusters or segments that are then used as inputs in subsequent analyses (for example, by segmenting potential customers into homogeneous subgroups).

Segmentation nodes

	The Auto Cluster node estimates and compares clustering models, which identify groups of records that have similar characteristics. The node works in the same manner as other automated modeling nodes, allowing you to experiment with multiple combinations of options in a single modeling pass. Models can be compared using basic measures with which to attempt to filter and rank the usefulness of the cluster models, and provide a measure based on the importance of particular fields.
	The K-Means node clusters the data set into distinct groups (or clusters). The method defines a fixed number of clusters, iteratively assigns records to clusters, and adjusts the cluster centers until further refinement can no longer improve the model. Instead of trying to predict an outcome, k-means uses a process known as unsupervised learning to uncover patterns in the set of input fields.
	The Kohonen node generates a type of neural network that can be used to cluster the data set into distinct groups. When the network is fully trained, records that are similar should be close together on the output map, while records that are different will be far apart. You can look at the number of observations captured by each unit in the model nugget to identify the strong units. This may give you a sense of the appropriate number of clusters.
	The TwoStep node uses a two-step clustering method. The first step makes a single pass through the data to compress the raw input data into a manageable set of subclusters. The second step uses a hierarchical clustering method to progressively

	merge the subclusters into larger and larger clusters. TwoStep has the advantage of automatically estimating the optimal number of clusters for the training data. It can handle mixed field types and large data sets efficiently.
	The Anomaly Detection node identifies unusual cases, or outliers, that do not conform to patterns of "normal" data. With this node, it is possible to identify outliers even if they do not fit any previously known patterns and even if you are not exactly sure what you are looking for.

Note: KNN, SVM, GENLin, Cox, SLRM, and Bayes Net nodes are part of the Extension Classification Module in SPSS Modeler and are supported under the Modeler Premium license.

In-Database Mining Models

IBM SPSS Modeler supports integration with data mining and modeling tools that are available from database vendors, including Oracle Data Miner and Microsoft Analysis Services. You can build, score, and store models inside the database—all from within the IBM SPSS Modeler application. For full details, see the *IBM SPSS Modeler In-Database Mining Guide*.

IBM SPSS Statistics Models

If you have a copy of IBM SPSS Statistics installed and licensed on your computer, you can access and run certain IBM SPSS Statistics routines from within IBM SPSS Modeler to build and score models.

Building Split Models

Split modeling enables you to use a single stream to build separate models for each possible value of a flag, nominal, or continuous input field, with the resulting models all being accessible from a single model nugget. The possible values for the input fields could have very different effects on the model. With split modeling, you can easily build the best-fitting model for each possible field value in a single execution of the stream.

Note that interactive modeling sessions cannot use splitting. With interactive modeling you specify each model individually, so there would be no advantage in using splitting, which builds multiple models automatically.

Split modeling works by designating a particular input field as a split field. You can do this by setting the field role to Split in the Type specification.

You can designate only fields with a measurement level of Flag, Nominal, Ordinal, or Continuous as split fields.

You can assign more than one input field as a split field. In this case, however, the number of models created can be greatly increased. A model is built for each possible combination of the values of the selected split fields. For example, if three input fields, each having three possible values, are designated as split fields, this will result in the creation of 27 different models.

Even after you assign one or more fields as split fields, you can still choose whether to create split models or a single model, by means of a check box setting on the modeling node dialog.

If split fields are defined but the check box is not selected, only a single model is generated. Likewise if the check box is selected but no split field is defined, splitting is ignored and a single model is generated.

When you run the stream, separate models are built behind the scenes for each possible value of the split field or fields, but only a single model nugget is placed in the models palette and the stream canvas. A split-model nugget is denoted by the split symbol; this is two gray rectangles overlaid on the nugget image.

When you browse the split-model nugget, you see a list of all the separate models that have been built.

You can investigate an individual model from a list by double-clicking its nugget icon in the viewer. Doing so opens a standard browser window for the individual model. When the nugget is on the canvas, double-clicking a graph thumbnail opens the full-size graph. See the topic [Split Model Viewer](#) for more information.

Once a model has been created as a split model, you cannot remove the split processing from it, nor can you undo splitting further downstream from a split-modeling node or nugget.

Example. A national retailer wants to estimate sales by product category at each of its stores around the country. Using split modeling, they designate the Store field of their input data as a split field, enabling them to build separate models for each category at each store in a single operation. They can then use the resulting information to control stock levels much more accurately than they could with only a single model.

- [Splitting and Partitioning](#)
- [Modeling nodes supporting split models](#)
- [Features Affected by Splitting](#)

Splitting and Partitioning

Splitting has some features in common with partitioning, but the two are used in very different ways.

Partitioning divides the dataset randomly into either two or three parts: training, testing and (optionally) validation, and is used to test the performance of a single model.

Splitting divides the dataset into as many parts as there are possible values for a split field, and is used to build multiple models.

Partitioning and splitting operate completely independently of each other. You can choose either, both or neither in a modeling node.

Modeling nodes supporting split models

A number of modeling nodes can create split models. The exceptions are Auto Cluster, PCA/Factor, Feature Selection, SLM, Random Trees, Tree-AS, Linear-AS, LSVM, the association models (Apriori, Carma and Sequence), the clustering models (K-Means, Kohonen, Two Step and Anomaly), Statistics models, and the nodes used for in-database modeling.

The modeling nodes that support split modeling are:

	C&R Tree		Bayes Net		Linear
	QUEST		GenLin		GLMM
	CHAID		KNN		STP
	C5.0		Cox		One-Class SVM
	Neural Net		Auto Classifier		XGBoost Tree
	Decision List		Auto Numeric		XGBoost Linear
	Regression		Logistic		HDBSCAN
	Discriminant		SVM		Time Series

Features Affected by Splitting

The use of split models affects a number of IBM® SPSS® Modeler features in various ways. This section provides guidance on using split models with other nodes in a stream.

Record Ops nodes

When you use split models in a stream that contains a Sample node, stratify records by the split field to achieve an even sampling of records. This option is available when you choose *Complex* as the sample method.

If the stream contains a Balance node, balancing applies to the overall set of input records, not to the subset of records inside a split.

When aggregating records by means of an Aggregate node, set the split fields to be key fields if you want to calculate aggregates for each split.

Field Ops nodes

The Type node is where you specify which field or fields to use as split fields.

Note: While the Ensemble node is used to combine two or more model nuggets, it cannot be used to reverse the action of splitting, as the split models are contained inside a single model nugget.

Modeling nodes

Split models do not support the calculation of predictor importance (the relative importance of the predictor input fields in estimating the model). Predictor importance settings are ignored when building split models.

Note: Adjusted propensity score settings are ignored when using a split model.

The KNN (nearest neighbor) node supports split models only if it is set to predict a target field. The alternative setting (only identify nearest neighbors) does not create a model. If the option Automatically select k is chosen, each of the split models might have a different number of nearest neighbors. Thus the overall model has a number of generated columns equal to the largest number of nearest neighbors that are found across all the split models. For those split models where the number of nearest neighbors is less than this maximum, there is a corresponding number of columns filled with `$null` values. See the topic [KNN node](#) for more information.

Database Modeling nodes

The in-database modeling nodes do not support split models.

Model nuggets

Export to PMML from a split model nugget is not possible, as the nugget contains multiple models and PMML does not support such a packaging. Export to text or HTML is possible.

Modeling Node Fields Options

All modeling nodes have a Fields tab, where you can specify the fields to be used in building the model.

Before you can build a model, you need to specify which fields you want to use as targets and as inputs. With a few exceptions, all modeling nodes will use field information from an upstream Type node. If you are using a Type node to select input and target fields, you don't need to change anything on this tab. (Exceptions include the Sequence node and the Text Extraction node, which require that field settings be specified in the modeling node.)

Use type node settings. This option tells the node to use field information from an upstream Type node. This is the default.

Use custom settings. This option tells the node to use field information specified here instead of that given in any upstream Type node(s). After selecting this option, specify the fields below as required.

Note: Not all fields are displayed for all nodes.

- **Use transactional format (Apriori, CARMA, MS Association Rules and Oracle Apriori nodes only).** Select this check box if the source data is in **transactional format**. Records in this format have two fields, one for an ID and one for content. Each record represents a single transaction or item, and associated items are linked by having the same ID. Deselect this box if the data is in **tabular format**, in which items are represented by separate flags, where each flag field represents the presence or absence of a specific item and each record represents a complete set of associated items. See the topic [Tabular versus Transactional Data](#) for more information.
 - ID. For transactional data, select an ID field from the list. Numeric or symbolic fields can be used as the ID field. Each unique value of this field should indicate a specific unit of analysis. For example, in a market basket application, each ID might represent a single customer. For a Web log analysis application, each ID might represent a computer (by IP address) or a user (by login data).
 - IDs are contiguous. (Apriori and CARMA nodes only) If your data are presorted so that all records with the same ID are grouped together in the data stream, select this option to speed up processing. If your data are not presorted (or you are not sure), leave this option unselected and the node will sort the data automatically.
Note: If your data are not sorted and you select this option, you may get invalid results in your model.
 - Content. Specify the content field(s) for the model. These fields contain the items of interest in association modeling. You can specify multiple flag fields (if data are in tabular format) or a single nominal field (if data are in transactional format).
- **Target.** For models that require one or more target fields, select the target field or fields. This is similar to setting the field role to *Target* in a Type node.
- **Evaluation.** (For Auto Cluster models only.) No target is specified for cluster models; however, you can select an evaluation field to identify its level of importance. In addition, you can evaluate how well the clusters differentiate values of this field, which in turn indicates whether the clusters can be used to predict this field. *Note* The evaluation field must be a string with more than one value.
 - Inputs. Select the input field or fields. This is similar to setting the field role to *Input* in a Type node.
 - Partition. This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to generate the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)

- **Splits.** For split models, select the split field or fields. This is similar to setting the field role to *Split* in a Type node. You can designate only fields with a measurement level of Flag, Nominal, Ordinal or Continuous as split fields. Fields chosen as split fields cannot be used as target, input, partition, frequency or weight fields. See the topic [Building Split Models](#) for more information.
- **Use frequency field.** This option enables you to select a field as a frequency weight. Use this if the records in your training data represent more than one unit each--for example, if you are using aggregated data. The field values should be the number of units represented by each record. See the topic [Using Frequency and Weight Fields](#) for more information.

Note: If you see the error message Metadata (on input/output fields) not valid, ensure that you have specified all fields that are required, such as the frequency field.

- **Use weight field.** This option enables you to select a field as a case weight. Case weights are used to account for differences in variance across levels of the output field. See the topic [Using Frequency and Weight Fields](#) for more information.
- **Consequents.** For rule induction nodes (Apriori), select the fields to be used as consequents in the resulting rule set. (This corresponds to fields with role *Target* or *Both* in a Type node.)
- **Antecedents.** For rule induction nodes (Apriori), select the fields to be used as antecedents in the resulting rule set. (This corresponds to fields with role *Input* or *Both* in a Type node.)

Some models have a Fields tab that differs from those described in this section.

- See the topic [Sequence Node Fields Options](#) for more information.
- See the topic [CARMA Node Fields Options](#) for more information.
- [Using Frequency and Weight Fields](#)

Related information

- [Overview of modeling nodes](#)

Using Frequency and Weight Fields

Frequency and weight fields are used to give extra importance to some records over others, for example, because you know that one section of the population is under-represented in the training data (weight) or because one record represents a number of identical cases (frequency).

- Values for a frequency field should be positive integers. Records with a negative or zero frequency weight are excluded from the analysis. Non-integer frequency weights are rounded to the nearest integer.
- Case weight values should be positive but need not be integer values. Records with a negative or zero case weight are excluded from the analysis.

Scoring Frequency and Weight Fields

Frequency and weight fields are used in training models, but are not used in scoring, because the score for each record is based on its characteristics regardless of how many cases it represents. For example, suppose you have the data in the following table.

Table 1. Data example

Married	Responded
Yes	Yes
Yes	Yes
Yes	Yes
Yes	No
No	Yes
No	No
No	No

Based on this, you conclude that three out of four married people respond to the promotion, and two out of three unmarried people didn't respond. So you will score any new records accordingly, as shown in the following table.

Table 2. Scored records example

Married	\$-Responded	\$RP-Responded
Yes	Yes	0.75 (three/four)
No	No	0.67 (two/three)

Alternatively, you could store your training data more compactly, using a frequency field, as shown in the following table.

Table 3. Scored records alternative example

Married	Responded	Frequency
---------	-----------	-----------

Married	Responded	Frequency
Yes	Yes	3
Yes	No	1
No	Yes	1
No	No	2

Since this represents exactly the same dataset, you will build the same model and predict responses based solely on marital status. If you have ten married people in your scoring data, you will predict Yes for each of them regardless of whether they are presented as ten separate records, or one with a frequency value of 10. Weight, although generally not an integer, can be thought of as similarly indicating the importance of a record. This is why frequency and weight fields are not used when scoring records.

Evaluating and Comparing Models

Some model types support frequency fields, some support weight fields, and some support both. But in all cases where they apply, they are used only for model building and are not considered when evaluating models using an Evaluation node or Analysis node, or when ranking models using most of the methods supported by the Auto Classifier and Auto Numeric nodes.

- When comparing models (using evaluation charts, for example), frequency and weight values will be ignored. This enables a level comparison between models that use these fields and models that don't, but means that for an accurate evaluation, a dataset that accurately represents the population without relying on a frequency or weight field must be used. In practical terms, you can do this by making sure that models are evaluated using a testing sample in which the value of the frequency or weight field is always null or 1. (This restriction only applies when evaluating models; if frequency or weight values were always 1 for both training and testing samples, there would be no reason to use these fields in the first place.)
- If using Auto Classifier, frequency can be taken into account if ranking models based on Profit, so this method is recommended in that case.
- If necessary, you can split the data into training and testing samples using a Partition node.

Related information

- [Automated Modeling Nodes](#)
-

Modeling Node Analyze Options

Many modeling nodes include an Analyze tab that enables you to obtain predictor importance information along with raw and adjusted propensity scores.

Model Evaluation

Calculate predictor importance. For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may take longer to calculate for some models, particularly when working with large datasets, and is off by default for some models as a result. Predictor importance is not available for decision list models. See [Predictor Importance](#) for more information.

Propensity Scores

Propensity scores can be enabled in the modeling node, and on the Settings tab in the model nugget. This functionality is available only when the selected target is a flag field. See the topic [Propensity Scores](#) for more information.

Calculate raw propensity scores. Raw propensity scores are derived from the model based on the training data only. If the model predicts the true value (will respond), then the propensity is the same as P, where P is the probability of the prediction. If the model predicts the false value, then the propensity is calculated as $(1 - P)$.

- If you choose this option when building the model, propensity scores will be enabled in the model nugget by default. However, you can always choose to enable raw propensity scores in the model nugget whether or not you select them in the modeling node.
- When scoring the model, raw propensity scores will be added in a field with the letters *RP* appended to the standard prefix. For example, if the predictions are in a field named *\$R-churn*, the name of the propensity score field will be *\$RRP-churn*.

Calculate adjusted propensity scores. Raw propensities are based purely on estimates given by the model, which may be overfitted, leading to over-optimistic estimates of propensity. Adjusted propensities attempt to compensate by looking at how the model performs on the test or validation partitions and adjusting the propensities to give a better estimate accordingly.

- This setting requires that a valid partition field is present in the stream.
- Unlike raw confidence scores, adjusted propensity scores must be calculated when building the model; otherwise, they will not be available when scoring the model nugget.
- When scoring the model, adjusted propensity scores will be added in a field with the letters *AP* appended to the standard prefix. For example, if the predictions are in a field named *\$R-churn*, the name of the propensity score field will be *\$RAP-churn*. Adjusted propensity scores are not available for logistic regression models.

- When calculating the adjusted propensity scores, the test or validation partition used for the calculation must not have been balanced. To avoid this, be sure the Only balance training data option is selected in any upstream Balance nodes. In addition, if a complex sample has been taken upstream this will invalidate the adjusted propensity scores.
- Adjusted propensity scores are not available for "boosted" tree and rule set models. See the topic [Boosted C5.0 Models](#) for more information.

Based on. For adjusted propensity scores to be computed, a partition field must be present in the stream. You can specify whether to use the testing or validation partition for this computation. For best results, the testing or validation partition should include at least as many records as the partition used to train the original model.

- [Propensity Scores](#)

Related information

- [Overview of modeling nodes](#)
-

Propensity Scores

For models that return a *yes* or *no* prediction, you can request propensity scores in addition to the standard prediction and confidence values. Propensity scores indicate the likelihood of a particular outcome or response. The following table contains an example.

Table 1. Propensity scores

Customer	Propensity to respond
Joe Smith	35%
Jane Smith	15%

Propensity scores are available only for models with flag targets, and indicate the likelihood of the *True* value defined for the field, as specified in a source or Type node.

Propensity Scores Versus Confidence Scores

Propensity scores differ from confidence scores, which apply to the current prediction, whether *yes* or *no*. In cases where the prediction is *no*, for example, a high confidence actually means a high likelihood *not* to respond. Propensity scores sidestep this limitation to enable easier comparison across all records. For example, a *no* prediction with a confidence of 0.85 translates to a raw propensity of 0.15 (or 1 minus 0.85).

Table 2. Confidence scores

Customer	Prediction	Confidence
Joe Smith	Will respond	.35
Jane Smith	Won't respond	.85

Obtaining Propensity Scores

- Propensity scores can be enabled on the Analyze tab in the modeling node or on the Settings tab in the model nugget. This functionality is available only when the selected target is a flag field. See the topic [Modeling Node Analyze Options](#) for more information.
- Propensity scores may also be calculated by the Ensemble node, depending on the ensemble method used.

Calculating Adjusted Propensity Scores

Adjusted propensity scores are calculated as part of the process of building the model, and will not be available otherwise. Once the model is built, it is then scored using data from the test or validation partition, and a new model to deliver adjusted propensity scores is constructed by analyzing the original model's performance on that partition. Depending on the type of model, one of two methods may be used to calculate the adjusted propensity scores.

- For rule set and tree models, adjusted propensity scores are generated by recalculating the frequency of each category at each tree node (for tree models) or the support and confidence of each rule (for rule set models). This results in a new rule set or tree model which is stored with the original model, to be used whenever adjusted propensity scores are requested. Each time the original model is applied to new data, the new model can subsequently be applied to the raw propensity scores to generate the adjusted scores.
- For other models, records produced by scoring the original model on the test or validation partition are then binned by their raw propensity score. Next, a neural network model is trained that defines a non-linear function that maps from the mean raw propensity in each bin to the mean observed propensity in the same bin. As noted earlier for tree models, the resulting neural net model is stored with the original model, and can be applied to the raw propensity scores whenever adjusted propensity scores are requested.

Caution regarding missing values in the testing partition. Handling of missing input values in the testing/validation partition varies by model (see individual model scoring algorithms for details). The C5 model cannot compute adjusted propensities when there are missing inputs.

Misclassification Costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

With the exception of C5.0 models, misclassification costs are not applied when scoring a model and are not taken into account when ranking or comparing models using an Auto Classifier node, evaluation chart, or Analysis node. A model that includes costs may not produce fewer errors than one that doesn't and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of less expensive errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select Use misclassification costs and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying A as B to be 2.0, the cost of misclassifying B as A will still have the default value of 1.0 unless you explicitly change it as well.

Note: Only the Decision Trees model allows costs to be specified at build time.

Model Nuggets

Figure 1. Model nugget



A model nugget is a container for a model, that is, the set of rules, formulas or equations that represent the results of your model building operations in SPSS® Modeler. The main purpose of a nugget is for scoring data to generate predictions, or to enable further analysis of the model properties. Opening a model nugget on the screen enables you to see various details about the model, such as the relative importance of the input fields in creating the model. To view the predictions, you need to attach and execute a further process or output node. See the topic [Using Model Nuggets in Streams](#) for more information.

Figure 2. Model link from modeling node to model nugget



When you successfully execute a modeling node, a corresponding model nugget is placed on the stream canvas, where it is represented by a gold, diamond-shaped icon (hence the name "nugget"). On the stream canvas, the nugget is shown with a connection (solid line) to the nearest suitable node before the modeling node, and a link (dotted line) to the modeling node itself.

The nugget is also placed in the Models palette in the upper right corner of the IBM® SPSS Modeler window. From either location, nuggets can be selected and browsed to view details of the model.

Nuggets are always placed in the Models palette when a modeling node is successfully executed. You can set a user option to control whether the nugget is additionally placed on the stream canvas.

The following topics provide information on using model nuggets in IBM SPSS Modeler. For an in-depth understanding of the algorithms used, see the *IBM SPSS Modeler Algorithms Guide*, available as a PDF file as part of your product download.

- [Model Links](#)
- [Replacing a model](#)
- [The models palette](#)
- [Browsing model nuggets](#)
- [Model Nugget Summary / Information](#)
- [Predictor Importance](#)
- [Ensemble Viewer](#)
- [Model Nuggets for Split Models](#)
- [Using Model Nuggets in Streams](#)
- [Regenerating a modeling node](#)
- [Importing and exporting models as PMML](#)
- [Publishing models for a scoring adapter](#)
- [Unrefined Models](#)

Model Links

Figure 1. Model link from modeling node to model nugget



By default, a nugget is shown on the canvas with a link to the modeling node that created it. This is especially useful in complex streams with several nuggets, enabling you to identify the nugget that will be updated by each modeling node. Each link contains a symbol to indicate whether the model is replaced when the modeling node is executed. See the topic [Replacing a model](#) for more information.

- [Defining and Removing Model Links](#)
- [Copying and Pasting Model Links](#)
- [Model Links and SuperNodes](#)

Defining and Removing Model Links

You can define and remove links manually on the canvas. When you are defining a new link, the cursor changes to the link cursor.

Figure 1. Link cursor



Defining a new link (context menu)

1. Right-click on the modeling node from which you want the link to start.
2. Choose Define Model Link from the context menu.
3. Click the nugget where you want the link to end.

Defining a new link (main menu)

1. Click the modeling node from which you want the link to start.
2. From the main menu, choose:
Edit > Node > Define Model Link
3. Click the nugget where you want the link to end.

Removing an existing link (context menu)

1. Right-click on the nugget at the end of the link.
2. Choose Remove Model Link from the context menu.

Alternatively:

1. Right-click on the symbol in the middle of the link.
2. Choose Remove Link from the context menu.

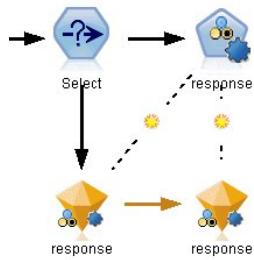
Removing an existing link (main menu)

1. Click the modeling node or nugget from which you want to remove the link.
2. From the main menu, choose:
Edit > Node > Remove Model Link

Copying and Pasting Model Links

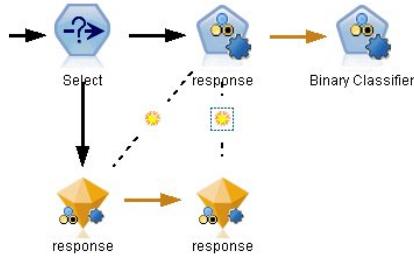
If you copy a linked nugget, without its modeling node, and paste it into the same stream, the nugget is pasted with a link to the modeling node. The new link has the same model replacement status (see [Replacing a model](#)) as the original link.

Figure 1. Copying and pasting a linked nugget



If you copy and paste a nugget together with its linked modeling node, the link is retained whether the objects are pasted into the same stream or a new stream.

Figure 2. Copying and pasting a linked nugget



Note: If you copy a linked nugget, without its modeling node, and paste the nugget into a new stream (or into a SuperNode that does not contain the modeling node), the link is broken and only the nugget is pasted.

Model Links and SuperNodes

If you define a SuperNode to include either the modeling node or the model nugget of a linked model (but not both), the link is broken. Expanding the SuperNode does not restore the link; you can only do this by undoing creation of the SuperNode.

Replacing a model

You can choose whether to replace (that is, update) an existing nugget on re-execution of the modeling node that created the nugget. If you turn off the replacement option, a new nugget is created when you re-execute the modeling node.

Each link from modeling node to nugget contains a symbol to indicate whether the model is replaced when the modeling node is re-executed.

Figure 1. Model link with model replacement turned on



The link is initially shown with model replacement turned on, depicted by the small sunburst symbol in the link. In this state, re-executing the modeling node at one end of the link simply updates the nugget at the other end.

Figure 2. Model link with model replacement turned off



If model replacement is turned off, the link symbol is replaced by a gray dot. In this state, re-executing the modeling node at one end of the link adds a new, updated version of the nugget to the canvas.

In either case, in the Models palette the existing nugget is updated or a new nugget is added, depending on the setting of the Replace previous model system option.

Order of execution

When you execute a stream with multiple branches containing model nuggets, the stream is first evaluated to make sure that a branch with model replacement turned on is executed before any branch that uses the resulting model nugget.

If your requirements are more complex, you can set the order of execution manually through scripting.

Changing the model replacement setting

1. Right-click the symbol on the link.
2. Choose Turn On(Off) Model Replacement as desired.

Note: The model replacement setting on a model link overrides the setting on the Notifications tab of the User Options dialog (Tools > Options > User Options).

The models palette

The models palette (on the Models tab in the managers window) enables you to use, examine, and modify model nuggets in various ways.

Figure 1. Models palette



Right-clicking a model nugget in the models palette opens a context menu with the following options:

- Add To Stream. Adds the model nugget to the currently active stream. If there is a selected node in the stream, the model nugget will be connected to the selected node when such a connection is possible, or otherwise to the nearest possible node. The nugget is displayed with a link to the modeling node that created the model, if that node is still in the stream.
- Browse. Opens the model browser for the nugget.
- Rename and Annotate. Allows you to rename the model nugget and/or modify the annotation for the nugget.
- Generate Modeling Node. If you have a model nugget that you want to modify or update and the stream used to create the model is not available, you can use this option to recreate a modeling node with the same options used to create the original model.
- Save Model, Save Model As. Saves the model nugget to an external generated model (.gm) binary file.
- Store Model. Stores the model nugget in IBM® SPSS® Collaboration and Deployment Services Repository.
- Export PMML. Exports the model nugget as predictive model markup language (PMML), which can be used for scoring new data outside of IBM SPSS Modeler. Export PMML is available for all generated model nodes.
- Add to Project. Saves the model nugget and adds it to the current project. On the Classes tab, the nugget will be added to the Generated Models folder. On the CRISP-DM tab, it will be added to the default project phase.
- Delete. Deletes the model nugget from the palette.

Right-clicking an unoccupied area in the models palette opens a context menu with the following options:

- Open Model. Loads a model nugget previously created in IBM SPSS Modeler.
- Retrieve Model. Retrieves a stored model from an IBM SPSS Collaboration and Deployment Services repository.
- Load Palette. Loads a saved models palette from an external file.
- Retrieve Palette. Retrieves a stored models palette from an IBM SPSS Collaboration and Deployment Services repository.
- Save Palette. Saves the entire contents of the models palette to an external generated models palette (.gen) file.
- Store Palette. Stores the entire contents of the models palette in an IBM SPSS Collaboration and Deployment Services repository.
- Clear Palette. Deletes all nuggets from the palette.
- Add Palette To Project. Saves the models palette and adds it to the current project. On the Classes tab, the nugget will be added to the Generated Models folder. On the CRISP-DM tab, it will be added to the default project phase.
- Import PMML. Loads a model from an external file. You can open, browse, and score PMML models created by IBM SPSS Statistics or other applications that support this format. See the topic [Importing and exporting models as PMML](#) for more information.

Browsing model nuggets

The model nugget browsers enable you to examine and use the results of your models. From the browser, you can save, print, or export the generated model, examine the model summary, and view or edit annotations for the model. For some types of model nugget, you can also generate new nodes, such as Filter nodes or Rule Set nodes. For some models, you can also view model parameters, such as rules or cluster centers. For some types of models (tree-based models and cluster models), you can view a graphical representation of the structure of the model. Controls for using the model nugget browsers are described below.

Menus

File menu. All model nuggets have a File menu, containing some subset of the following options:

- Save Node. Saves the model nugget to a node (.nod) file.
- Store Node. Stores the model nugget in an IBM® SPSS® Collaboration and Deployment Services repository.
- Header and Footer. Allows you to edit the page header and footer for printing from the nugget.
- Page Setup. Allows you to change the page setup for printing from the nugget.
- Print Preview. Displays a preview of how the nugget will look when printed. Select the information you want to preview from the submenu.
- Print. Prints the contents of the nugget. Select the information you want to print from the submenu.
- Print View. Prints the current view or all views.
- Export Text. Exports the contents of the nugget to a text file. Select the information you want to export from the submenu.
- Export HTML. Exports the contents of the nugget to an HTML file. Select the information you want to export from the submenu.
- Export PMML. Exports the model as predictive model markup language (PMML), which can be used with other PMML-compatible software. See the topic [Importing and exporting models as PMML](#) for more information.
- Export SQL. Exports the model as structured query language (SQL), which can be edited and used with other databases.
Note: SQL Export is available only from the following models: C5, C&RT, CHAID, QUEST, Linear Regression, Logistic Regression, Neural Net, PCA/Factor, and Decision List models.
- Publish for Server Scoring Adapter. Publishes the model to a database that has a scoring adapter installed, enabling model scoring to be performed within the database. See the topic [Publishing models for a scoring adapter](#) for more information.

Generate menu. Most model nuggets also have a Generate menu, enabling you to generate new nodes based on the model nugget. The options available from this menu will depend on the type of model you are browsing. See the specific model nugget type for details about what you can generate from a particular model.

View menu. On the Model tab of a nugget, this menu enables you to display or hide the various visualization toolbars that are available in the current mode. To make the full set of toolbars available, select Edit Mode (the paintbrush icon) from the General toolbar.

Preview button. Some model nuggets have a Preview button, which enables you to see a sample of the model data, including the extra fields created by the modeling process. The default number of rows displayed is 10; however, you can change this in the stream properties.

Add to Current Project button. Saves the model nugget and adds it to the current project. On the Classes tab, the nugget will be added to the Generated Models folder. On the CRISP-DM tab, it will be added to the default project phase.

Model Nugget Summary / Information

The Summary tab or Information view for a model nugget displays information about the fields, build settings, and model estimation process. Results are presented in a tree view that can be expanded or collapsed by clicking specific items.

Analysis. Displays information about the model. Specific details vary by model type, and are covered in the section for each model nugget. In addition, if you execute an Analysis node that is attached to this modeling node, information from that analysis is also displayed in this section.

Fields. Lists the fields that are used as the target and the inputs in building the model. For split models, also lists the fields that determined the splits.

Note: In the Information view for neural networks, linear, and other models with either boosting or bagging modes, the icon that is shown is the same (nominal icon), regardless of whether the type is a flag, nominal, or ordinal.

Build Settings / Options. Contains information about the settings that are used in building the model.

Training Summary. Shows the type of model, the stream that is used to create it, the user who created it, when it was built, and the elapsed time for building the model. Note that the elapsed time for building the model is only available on the Summary tab, not the Information view.

Related information

- [Model Nuggets](#)
- [The models palette](#)
- [Browsing model nuggets](#)
- [Predictor Importance](#)
- [Using Model Nuggets in Streams](#)

Predictor Importance

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

Predictor importance is available for models that produce an appropriate statistical measure of importance, including neural networks, decision trees (C&R Tree, C5.0, CHAID, and QUEST), Bayesian networks, discriminant, SVM, and SLM models, linear and logistic regression, generalized linear, and nearest neighbor (KNN) models. For most of these models, predictor importance can be enabled on the Analyze tab in the modeling node. See the topic [Modeling Node Analyze Options](#) for more information. For KNN models, see [Neighbors](#).

Note: Predictor importance is not supported for split models. Predictor importance settings are ignored when building split models. See the topic [Building Split Models](#) for more information.

Calculating predictor importance may take significantly longer than model building, particularly when using large datasets. It takes longer to calculate for SVM and logistic regression than for other models, and is disabled for these models by default. If using a dataset with a large number of predictors, initial screening using a Feature Selection node may give faster results (see below).

- Predictor importance is calculated from the test partition, if available. Otherwise the training data is used.
- For SLM models, predictor importance is available but is computed by the SLM algorithm. See the topic [SLM Model Nuggets](#) for more information.
- You can use IBM® SPSS® Modeler's graph tools to interact, edit, and save the graph.
- Optionally, you can generate a Filter node based on the information in the predictor importance chart. See the topic [Filtering Variables Based on Importance](#) for more information.

Predictor Importance and Feature Selection

The predictor importance chart displayed in a model nugget may seem to give results similar to the Feature Selection node in some cases. While feature selection ranks each input field based on the strength of its relationship to the specified target, independent of other inputs, the predictor importance chart indicates the relative importance of each input for *this* particular model. Thus feature selection will be more conservative in screening inputs. For example, if *job title* and *job category* are both strongly related to salary, then feature selection would indicate that both are important. But in modeling, interactions and correlations are also taken into consideration. Thus you might find that only one of two inputs is used if both duplicate much of the same information. In practice, feature selection is most useful for preliminary screening, particularly when dealing with large datasets with large numbers of variables, and predictor importance is more useful in fine-tuning the model.

Predictor Importance Differences Between Single Models and Automated Modeling Nodes

Depending on whether you are creating a single model from an individual node, or using an automated modelling node to produce results, you may see slight differences in the predictor importance. Such differences in implementation are due to some engineering restrictions.

For example, with single classifiers such as CHAID the calculation applies a stopping rule and uses probability values when computing importance values. In contrast, the Auto Classifier does not use a stopping rule and uses predicted labels directly in the calculation. These differences can mean that if you produce a single model using Auto Classifier, the importance value can be considered as a rough estimation, compared with that computed for a single classifier. To obtain the most accurate predictor importance values we suggest using a single node instead of the automated modelling nodes.

- [Filtering Variables Based on Importance](#)

Related information

- [Overview of modeling nodes](#)
- [Filtering Variables Based on Importance](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Browsing model nuggets](#)
- [Model Nugget Summary / Information](#)
- [Using Model Nuggets in Streams](#)

Filtering Variables Based on Importance

Optionally, you can generate a Filter node based on the information in the predictor importance chart.

Mark the predictors you want to include on the chart, if applicable, and from the menus choose:

Generate...>[Filter Node \(Predictor Importance\)](#)

OR

...Field Selection (Predictor Importance)

Top number of variables. Includes or excludes the most important predictors up to the specified number.

Importance greater than. Includes or excludes all predictors with relative importance greater than the specified value.

Ensemble Viewer

- [Models for ensembles](#)
-

Models for ensembles

The model for an ensemble provides information about the component models in the ensemble and the performance of the ensemble as a whole.

The main (view-independent) toolbar allows you to choose whether to use the ensemble or a reference model for scoring. If the ensemble is used for scoring you can also select the combining rule. These changes do not require model re-execution; however, these choices are saved to the model (nugget) for scoring and/or downstream model evaluation. They also affect PMML exported from the ensemble viewer.

Combining Rule. When scoring an ensemble, this is the rule used to combine the predicted values from the base models to compute the ensemble score value.

- Ensemble predicted values for **categorical** targets can be combined using voting, highest probability, or highest mean probability. **Voting** selects the category that has the highest probability most often across the base models. **Highest probability** selects the category that achieves the single highest probability across all base models. **Highest mean probability** selects the category with the highest value when the category probabilities are averaged across base models.
- Ensemble predicted values for **continuous** targets can be combined using the mean or median of the predicted values from the base models.

The default is taken from the specifications made during model building. Changing the combining rule recomputes the model accuracy and updates all views of model accuracy. The Predictor Importance chart also updates. This control is disabled if the reference model is selected for scoring.

Show All Combining rules. When selected, results for all available combining rules are shown in the model quality chart. The Component Model Accuracy chart is also updated to show reference lines for each voting method.

- [Model Summary \(ensemble viewer\)](#)
 - [Predictor Importance \(ensemble viewer\)](#)
 - [Predictor Frequency \(ensemble viewer\)](#)
 - [Component Model Accuracy \(ensemble viewer\)](#)
 - [Component Model Details \(ensemble viewer\)](#)
 - [Automatic Data Preparation \(ensemble viewer\)](#)
-

Model Summary (ensemble viewer)

The Model Summary view is a snapshot, at-a-glance summary of the ensemble quality and diversity.

Quality. The chart displays the accuracy of the final model, compared to a reference model and a naive model. Accuracy is presented in larger is better format; the "best" model will have the highest accuracy. For a categorical target, accuracy is simply the percentage of records for which the predicted value matches the observed value. For a continuous target, accuracy is 1 minus the ratio of the mean absolute error in prediction (the average of the absolute values of the predicted values minus the observed values) to the range of predicted values (the maximum predicted value minus the minimum predicted value).

For bagging ensembles, the reference model is a standard model built on the whole training partition. For boosted ensembles, the reference model is the first component model.

The naive model represents the accuracy if no model were built, and assigns all records to the modal category. The naive model is not computed for continuous targets.

Diversity. The chart displays the "diversity of opinion" among the component models used to build the ensemble, presented in larger is more diverse format. It is a measure of how much predictions vary across the base models. Diversity is not available for boosted ensemble models, nor is it shown for continuous targets.

Predictor Importance (ensemble viewer)

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

Predictor importance is not available for all ensemble models. The predictor set may vary across component models, but importance can be computed for predictors used in at least one component model.

Predictor Frequency (ensemble viewer)

The predictor set can vary across component models due to the choice of modeling method or predictor selection. The Predictor Frequency plot is a dot plot that shows the distribution of predictors across component models in the ensemble. Each dot represents one or more component models containing the predictor. Predictors are plotted on the y-axis, and are sorted in descending order of frequency; thus the topmost predictor is the one that is used in the greatest number of component models and the bottommost one is the one that was used in the fewest. The top 10 predictors are shown.

Predictors that appear most frequently are typically the most important. This plot is not useful for methods in which the predictor set cannot vary across component models.

Component Model Accuracy (ensemble viewer)

The chart is a dot plot of predictive accuracy for component models. Each dot represents one or more component models with the level of accuracy plotted on the y-axis. Hover over any dot to obtain information on the corresponding individual component model.

Reference lines. The plot displays color coded lines for the ensemble as well as the reference model and naïve models. A checkmark appears next to the line corresponding to the model that will be used for scoring.

Interactivity. The chart updates if you change the combining rule.

Boosted ensembles. A line chart is displayed for boosted ensembles.

Component Model Details (ensemble viewer)

The table displays information on component models, listed by row. By default, component models are sorted in ascending model number order. You can sort the rows in ascending or descending order by the values of any column.

Model. A number representing the sequential order in which the component model was created.

Accuracy. Overall accuracy formatted as a percentage.

Method. The modeling method.

Predictors. The number of predictors used in the component model.

Model Size. Model size depends on the modeling method: for trees, it is the number of nodes in the tree; for linear models, it is the number of coefficients; for neural networks, it is the number of synapses.

Records. The weighted number of input records in the training sample.

Automatic Data Preparation (ensemble viewer)

This view shows information about which fields were excluded and how transformed fields were derived in the automatic data preparation (ADP) step. For each field that was transformed or excluded, the table lists the field name, its role in the analysis, and the action taken by the ADP step. Fields are sorted by ascending alphabetical order of field names.

The action Trim outliers, if shown, indicates that values of continuous predictors that lie beyond a cutoff value (3 standard deviations from the mean) have been set to the cutoff value.

Model Nuggets for Split Models

The model nugget for a split model provides access to all the separate models created by the splits.

A split-model nugget contains:

- a list of all the split models created, together with a set of statistics about each model
- information about the overall model

From the list of split models, you can open up individual models to examine them further.

- [Split Model Viewer](#)

Split Model Viewer

The Model tab lists all the models contained in the nugget, and provides statistics in various forms about the split models. It has two general forms, depending on the modeling node.

Sort by. Use this list to choose the order in which the models are listed. You can sort the list based on the values of any of the display columns, in ascending or descending order. Alternatively, click on a column heading to sort the list by that column. Default is descending order of overall accuracy.

Show/hide columns menu. Click this button to display a menu from where you can choose individual columns to show or hide.

View. If you are using partitioning, you can choose to view the results for either the training data or the testing data.

For each split, the details shown are as follows:

Graph. A thumbnail indicating the data distribution for this model. When the nugget is on the canvas, double-click the thumbnail to open the full-size graph.

Model. An icon of the model type. Double-click the icon to open the model nugget for this particular split.

Split fields. The fields designated in the modeling node as split fields, with their various possible values.

No. Records in Split. The number of records involved in this particular split.

No. Fields Used. Ranks split models based on the number of input fields used.

Overall Accuracy (%). The percentage of records that is correctly predicted by the split model relative to the total number of records in that split.

Split. The column heading shows the field(s) used to create splits, and the cells are the split values. Double-click any split to open a Model Viewer for the model built for that split.

Accuracy. Overall accuracy formatted as a percentage.

Model Size. Model size depends on the modeling method: for trees, it is the number of nodes in the tree; for linear models, it is the number of coefficients; for neural networks, it is the number of synapses.

Records. The weighted number of input records in the training sample.

Using Model Nuggets in Streams

Model nuggets are placed in streams to enable you to score new data and generate new nodes. **Scoring** data enables you to use the information gained from model building to create predictions for new records. To see the results of scoring, you need to attach a terminal node (that is, a processing or output node) to the nugget and execute the terminal node.

For some models, model nuggets can also give you additional information about the quality of the prediction, such as confidence values or distances from cluster centers. Generating new nodes enables you to easily create new nodes based on the structure of the generated model. For example, most models that perform input field selection enable you to generate Filter nodes that will pass only input fields that the model identified as important.

Note: There can be small differences in the scores assigned to a given case by a given model when executed in different versions of IBM® SPSS® Modeler. This is usually a result of enhancements to the software between versions.

To Use a Model Nugget for Scoring Data

1. Connect the model nugget to a data source or stream that will pass data to it.
2. Add or connect one or more processing or output nodes (such as a Table or Analysis node) to the model nugget.
3. Execute one of the nodes downstream from the model nugget.

Note: You cannot use the Unrefined Rule node for scoring data. To score data based on an association rule model, use the Unrefined Rule node to generate a Rule Set nugget, and use the Rule Set nugget for scoring. See the topic [Generating a Rule Set from an Association Model Nugget](#) for more information.

To Use a Model Nugget for Generating Processing Nodes

1. On the palette, browse the model, or, on the stream canvas, edit the model.
2. Select the desired node type from the Generate menu of the model nugget browser window. The options available will vary, depending on the type of model nugget. See the specific model nugget type for details about what you can generate from a particular model.

Related information

- [Model Nuggets](#)
 - [The models palette](#)
 - [Browsing model nuggets](#)
 - [Model Nugget Summary / Information](#)
 - [Predictor Importance](#)
-

Regenerating a modeling node

If you have a model nugget that you want to modify or update and the stream used to create the model is not available, you can regenerate a modeling node with the same options used to create the original model.

To rebuild a model, right-click on the model in the models palette and choose Generate Modeling Node.

Alternatively, when browsing any model, choose Generate Modeling Node from the Generate menu.

The regenerated modeling node should be functionally identical to the one used to create the original model in most cases.

- For Decision Tree models, additional settings specified during the interactive session may also be stored with the node, and the Use tree directives option will be enabled in the regenerated modeling node.
 - For Decision List models, the Use saved interactive session information option will be enabled. See the topic [Decision List Model Options](#) for more information.
 - For Time Series models, the Continue estimation using existing model(s) option is enabled, which enables you to regenerate the previous model with current data. See the topic [Time Series Model Options](#) for more information.
-

Importing and exporting models as PMML

PMML, or predictive model markup language, is an XML format for describing data mining and statistical models, including inputs to the models, transformations used to prepare data for data mining, and the parameters that define the models themselves. IBM® SPSS® Modeler can import and export PMML, making it possible to share models with other applications that support this format, such as IBM SPSS Statistics.

For more information about PMML, see the Data Mining Group website (<http://www.dmg.org>).

To export a model

PMML export is supported for most of the model types generated in IBM SPSS Modeler. See the topic [Model types supporting PMML](#) for more information.

1. Right-click a model nugget on the models palette. (Alternatively, double-click a model nugget on the canvas and select the File menu.)
2. On the menu, click Export PMML.
3. In the Export (or Save) dialog box, specify a target directory and a unique name for the model.

Note:

You can change options for PMML export in the User Options dialog box. On the main menu, click:

Tools > Options > User Options

and click the PMML tab.

See the topic [Setting PMML Export Options](#) for more information.

To import a model saved as PMML

Models exported as PMML from IBM SPSS Modeler or another application can be imported into the models palette. See the topic [Model types supporting PMML](#) for more information.

1. In the models palette, right-click the palette and select Import PMML from the menu.
2. Select the file to import and specify options for variable labels as required.
3. Click Open.

Use variable labels if present in model. The PMML may specify both variable names and variable labels (such as Referrer ID for *RefID*) for variables in the data dictionary. Select this option to use variable labels if they are present in the originally exported PMML.

If you have selected the variable label option but there are no variable labels in the PMML, the variable names are used as normal.

- [Model types supporting PMML](#)
-

Model types supporting PMML

PMML Export

IBM SPSS Modeler models. The following models created in IBM® SPSS® Modeler can be exported as PMML 4.3:

- C&R Tree
- QUEST
- CHAID
- Neural Net
- C5.0
- Logistic Regression
- Genlin
- SVM
- Apriori
- Carma
- K-Means
- Kohonen
- TwoStep
- TwoStep-AS
- GLMM (PMML is exported for all GLMM models, but the PMML has only fixed effects)
- Decision List
- Cox
- Sequence (scoring for Sequence PMML models is not supported)
- Random Trees
- Tree-AS
- Linear
- Linear-AS
- Regression
- Logistic
- GLE
- LSVM
- KNN
- Association Rules

Database native models. For models generated using database-native algorithms, PMML export is not available. Models created using Analysis Services from Microsoft or Oracle Data Miner cannot be exported.

PMML Import

IBM SPSS Modeler can import and score PMML models generated by current versions of all IBM SPSS Statistics products, including models exported from IBM SPSS Modeler as well as model or transformation PMML generated by IBM SPSS Statistics 17.0 or later. Essentially, this means any PMML that the scoring engine can score, with the following exceptions:

- Apriori, CARMA, Anomaly Detection, Sequence, and Association Rules models cannot be imported.

- PMML models may not be browsed after importing into IBM SPSS Modeler even though they can be used in scoring. (Note that this includes models that were exported from IBM SPSS Modeler to begin with. To avoid this limitation, export the model as a generated model file [*.gm] rather than PMML.)
- Limited validation occurs on import, but full validation is performed on attempting to score the model. Thus it is possible for import to succeed but scoring to fail or produce incorrect results.

Note: For third party PMML imported into IBM SPSS Modeler, IBM SPSS Modeler will attempt to score valid PMML that can be recognized and scored. But it is not guaranteed that all PMML will score or that it will score in the same way as the application that generated it.

Publishing models for a scoring adapter

You can publish models to a database server that has a scoring adapter installed. A scoring adapter enables model scoring to be performed within the database by using the user-defined function (UDF) capabilities of the database. Performing scoring in the database avoids the need to extract the data before scoring. Publishing to a scoring adapter also generates some example SQL to execute the UDF.

To publish a scoring adapter

1. Double-click the model nugget to open it.
2. From the model nugget menu, choose:
File > Publish for Server Scoring Adapter
3. Fill in the relevant fields on the dialog box and click OK.

Database connection. The connection details for the database you want to use for the model.

Publish ID. (Db2 for z/OS databases only) An identifier for the model. If you rebuild the same model and use the same publish ID, the generated SQL remains the same, so it is possible to rebuild a model without having to change the application that uses the SQL previously generated. (For other databases the generated SQL is unique to the model.)

Generate Example SQL. If selected, generates the example SQL into the file specified in the File field.

Unrefined Models

An unrefined model contains information extracted from the data but is not designed for generating predictions directly. This means that it cannot be added to streams. Unrefined models are displayed as “diamonds in the rough” on the generated models palette.

Figure 1. Unrefined model icon



To see information about the unrefined rule model, right-click the model and choose Browse from the context menu. Like other models generated in IBM® SPSS® Modeler, the various tabs provide summary and rule information about the model created.

Generating nodes. The Generate menu enables you to create new nodes based on the rules.

- **Select Node.** Generates a Select node to select records to which the currently selected rule applies. This option is disabled if no rule is selected.
- **Rule set.** Generates a Rule Set node to predict values for a single target field. See the topic [Generating a Rule Set from an Association Model Nugget](#) for more information.

Related information

- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)

Generated Statistical Models Advanced Output

The Advanced tab on Linear Regression, Logistic Regression and Factor/PCA model nugget browsers displays detailed statistical information about the model. For more information about interpreting these model statistics, select the link for the specific model type below.

Related information

- [Logistic Model Nugget Advanced Output](#)
 - [Regression Model Nugget Advanced Output](#)
 - [PCA/Factor Model Nugget Advanced Output](#)
-

Cluster Models Model Tab

The Model tab for cluster models (TwoStep, K-Means, and Kohonen) contains detailed information about the clusters defined by the model.

When you first browse a cluster model nugget, the Model tab results are collapsed. To see the results of interest, use the expander control to the left of the item to unfold it, or click the Expand All button to show all results. To hide the results when finished viewing them, use the expander control to collapse the specific results that you want to hide, or click the Collapse All button to collapse all results.

Clusters. Clusters are labeled and the number of records assigned to each cluster is shown. Each cluster is described by its center, which can be thought of as the **prototype** for the cluster. The details available for the clusters depend on the type of cluster model. Select the specific type of cluster model below for more information.

Generated Rule Set/Decision Tree Model Tab

On the generated Rule Set node Model tab, you will see a list of rules extracted from the data by the algorithm. The format of the rules depends on the type of model generated. Select the specific model type below for details.

Related information

- [Rule Set Model Tab](#)
 - [Decision Tree Model Rules](#)
 - [Boosted C5.0 Models](#)
-

Screening Models

- [Screening Fields and Records](#)
 - [Feature Selection node](#)
 - [Feature Selection Model Nuggets](#)
 - [Anomaly Detection Node](#)
 - [Anomaly Detection Model Nuggets](#)
-

Screening Fields and Records

Several modeling nodes can be used during the preliminary stages of an analysis in order to locate fields and records that are most likely to be of interest in modeling. You can use the Feature Selection node to screen and rank fields by importance and the Anomaly Detection node to locate unusual records that do not conform to the known patterns of "normal" data.

	The Feature Selection node screens input fields for removal based on a set of criteria (such as the percentage of missing values); it then ranks the importance of remaining inputs relative to a specified target. For example, given a data set with hundreds of potential inputs, which are most likely to be useful in modeling patient outcomes?
	The Anomaly Detection node identifies unusual cases, or outliers, that do not conform to patterns of "normal" data. With this node, it is possible to identify outliers even if they do not fit any previously known patterns and even if you are not exactly sure what you are looking for.

Note that anomaly detection identifies unusual records or cases through cluster analysis based on the set of fields selected in the model without regard for any specific target (dependent) field and regardless of whether those fields are relevant to the pattern you are trying to predict. For this reason, you may want to use anomaly detection in combination with feature selection or another technique for screening and ranking fields. For example, you can use feature selection to identify the most important fields relative to a specific target and then use anomaly detection to locate the records that are the most unusual with respect to those fields. (An alternative approach would be to build a decision tree model and then examine any misclassified records as potential anomalies. However, this method would be more difficult to replicate or automate on a large scale.)

Related information

- [Anomaly Detection Node](#)
 - [Neural Net Node](#)
 - [Statistical Models](#)
 - [Clustering models](#)
 - [Association Rules](#)
 - [Time Series Node \(deprecated\)](#)
-

Feature Selection node

Data mining problems may involve hundreds, or even thousands, of fields that can potentially be used as inputs. As a result, a great deal of time and effort may be spent examining which fields or variables to include in the model. To narrow down the choices, the Feature Selection algorithm can be used to identify the fields that are most important for a given analysis. For example, if you are trying to predict patient outcomes based on a number of factors, which factors are the most likely to be important?

Feature selection consists of three steps:

- Screening. Removes unimportant and problematic inputs and records, or cases such as input fields with too many missing values or with too much or too little variation to be useful.
- Ranking. Sorts remaining inputs and assigns ranks based on importance.
- Selecting. Identifies the subset of features to use in subsequent models—for example, by preserving only the most important inputs and filtering or excluding all others.

In an age where many organizations are overloaded with too much data, the benefits of feature selection in simplifying and speeding the modeling process can be substantial. By focusing attention quickly on the fields that matter most, you can reduce the amount of computation required; more easily locate small but important relationships that might otherwise be overlooked; and, ultimately, obtain simpler, more accurate, and more easily explainable models. By reducing the number of fields used in the model, you may find that you can reduce scoring times as well as the amount of data collected in future iterations.

Example. A telephone company has a data warehouse containing information about responses to a special promotion by 5,000 of the company's customers. The data includes a large number of fields containing customers' ages, employment, income, and telephone usage statistics. Three target fields show whether or not the customer responded to each of three offers. The company wants to use this data to help predict which customers are most likely to respond to similar offers in the future.

Requirements. A single target field (one with its role set to *Target*), along with multiple input fields that you want to screen or rank relative to the target. Both target and input fields can have a measurement level of *Continuous* (numeric range) or *Categorical*.

- [Feature Selection Model Settings](#)
 - [Feature Selection Options](#)
-

Feature Selection Model Settings

The settings on the Model tab include standard model options along with settings that enable you to fine-tune the criteria for screening input fields.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Screening Input Fields

Screening involves removing inputs or cases that do not add any useful information with respect to the input/target relationship. Screening options are based on attributes of the field in question without respect to predictive power relative to the selected target field. Screened fields are excluded from the computations used to rank inputs and optionally can be filtered or removed from the data used in modeling.

Fields can be screened based on the following criteria:

- **Maximum percentage of missing values.** Screens fields with too many missing values, expressed as a percentage of the total number of records. Fields with a large percentage of missing values provide little predictive information.
- **Maximum percentage of records in a single category.** Screens fields that have too many records falling into the same category relative to the total number of records. For example, if 95% of the customers in the database drive the same type of car, including this information is not useful in distinguishing one customer from the next. Any fields that exceed the specified maximum are screened. This option applies to categorical fields only.
- **Maximum number of categories as a percentage of records.** Screens fields with too many categories relative to the total number of records. If a high percentage of the categories contains only a single case, the field may be of limited use. For example, if every customer

wears a different hat, this information is unlikely to be useful in modeling patterns of behavior. This option applies to categorical fields only.

- **Minimum coefficient of variation.** Screens fields with a coefficient of variance less than or equal to the specified minimum. This measure is the ratio of the input field standard deviation to the mean of the input field. If this value is near zero, there is not much variability in the values for the variable. This option applies to continuous (numeric range) fields only.
- **Minimum standard deviation.** Screens fields with standard deviation less than or equal to the specified minimum. This option applies to continuous (numeric range) fields only.

Records with missing data. Records or cases that have missing values for the target field, or missing values for all inputs, are automatically excluded from all computations used in the rankings.

Related information

- [Feature Selection node](#)
 - [Feature Selection Options](#)
 - [Feature Selection Model Nuggets](#)
 - [Feature Selection Model Results](#)
 - [Selecting Fields by Importance](#)
 - [Generating a Filter from a Feature Selection Model](#)
-

Feature Selection Options

The Options tab allows you to specify the default settings for selecting or excluding input fields in the model nugget. You can then add the model to a stream to select a subset of fields for use in subsequent model-building efforts. Alternatively, you can override these settings by selecting or deselecting additional fields in the model browser after generating the model. However, the default settings make it possible to apply the model nugget without further changes, which may be particularly useful for scripting purposes.

See the topic [Feature Selection Model Results](#) for more information.

The following options are available:

All fields ranked. Selects fields based on their ranking as *important*, *marginal*, or *unimportant*. You can edit the label for each ranking as well as the cutoff values used to assign records to one rank or another.

Top number of fields. Selects the top n fields based on importance.

Importance greater than. Selects all fields with importance greater than the specified value.

The target field is always preserved regardless of the selection.

Importance Ranking Options

All categorical. When all inputs and the target are categorical, importance can be ranked based on any of four measures:

- **Pearson chi-square.** Tests for independence of the target and the input without indicating the strength or direction of any existing relationship.
- **Likelihood-ratio chi-square.** Similar to Pearson's chi-square but also tests for target-input independence.
- **Cramer's V.** A measure of association based on Pearson's chi-square statistic. Values range from 0, which indicates no association, to 1, which indicates perfect association.
- **Lambda.** A measure of association reflecting the proportional reduction in error when the variable is used to predict the target value. A value of 1 indicates that the input field perfectly predicts the target, while a value of 0 means the input provides no useful information about the target.

Some categorical. When some—but not all—inputs are categorical and the target is also categorical, importance can be ranked based on either the Pearson or likelihood-ratio chi-square. (Cramer's V and lambda are not available unless all inputs are categorical.)

Categorical versus continuous. When ranking a categorical input against a continuous target or vice versa (one or the other is categorical but not both), the F statistic is used.

Both continuous. When ranking a continuous input against a continuous target, the t statistic based on the correlation coefficient is used.

Related information

- [Feature Selection node](#)
 - [Feature Selection Model Settings](#)
 - [Feature Selection Model Nuggets](#)
 - [Feature Selection Model Results](#)
 - [Selecting Fields by Importance](#)
-

- [Generating a Filter from a Feature Selection Model](#)
-

Feature Selection Model Nuggets

Feature Selection model nuggets display the importance of each input relative to a selected target, as ranked by the Feature Selection node. Any fields that were screened out prior to the ranking are also listed. See the topic [Feature Selection node](#) for more information.

When you run a stream containing a Feature Selection model nugget, the model acts as a filter that preserves only selected inputs, as indicated by the current selection on the Model tab. For example, you could select all fields ranked as important (one of the default options) or manually select a subset of fields on the Model tab. The target field is also preserved regardless of the selection. All other fields are excluded.

Filtering is based on the field name only; for example, if you select *age* and *income*, any field that matches either of these names will be preserved. The model does not update field rankings based on new data; it simply filters fields based on the selected names. For this reason, care should be used in applying the model to new or updated data. When in doubt, regenerating the model is recommended.

- [Feature Selection Model Results](#)
- [Selecting Fields by Importance](#)
- [Generating a Filter from a Feature Selection Model](#)

Related information

- [Feature Selection node](#)
 - [Feature Selection Model Settings](#)
 - [Feature Selection Options](#)
 - [Feature Selection Model Results](#)
 - [Selecting Fields by Importance](#)
 - [Generating a Filter from a Feature Selection Model](#)
-

Feature Selection Model Results

The Model tab for a Feature Selection model nugget displays the rank and importance of all inputs in the upper pane and enables you to select fields for filtering by using the check boxes in the column on the left. When you run the stream, only the selected fields are preserved; the other fields are discarded. The default selections are based on the options specified in the model-building node, but you can select or deselect additional fields as needed.

The lower pane lists inputs that have been excluded from the rankings based on the percentage of missing values or on other criteria specified in the modeling node. As with the ranked fields, you can choose to include or discard these fields by using the check boxes in the column on the left. See the topic [Feature Selection Model Settings](#) for more information.

- To sort the list by rank, field name, importance, or any of the other displayed columns, click on the column header. Or, to use the toolbar, select the desired item from the Sort By list, and use the up and down arrows to change the direction of the sort.
- You can use the toolbar to check or uncheck all fields and to access the Check Fields dialog box, which enables you to select fields by rank or importance. You can also press the Shift and Ctrl keys while clicking on fields to extend the selection and use the spacebar to toggle on or off a group of selected fields. See the topic [Selecting Fields by Importance](#) for more information.
- The threshold values for ranking inputs as important, marginal, or unimportant are displayed in the legend below the table. These values are specified in the modeling node. See the topic [Feature Selection Options](#) for more information.

Related information

- [Feature Selection node](#)
 - [Feature Selection Model Settings](#)
 - [Feature Selection Options](#)
 - [Feature Selection Model Nuggets](#)
 - [Selecting Fields by Importance](#)
 - [Generating a Filter from a Feature Selection Model](#)
-

Selecting Fields by Importance

When scoring data using a Feature Selection model nugget, all fields selected from the list of ranked or screened fields--as indicated by the check boxes in the column on the left--will be preserved. Other fields will be discarded. To change the selection, you can use the toolbar to

access the Check Fields dialog box, which enables you to select fields by rank or importance.

All fields marked. Selects all fields marked as important, marginal, or unimportant.

Top number of fields. Allows you to select the top n fields based on importance.

Importance greater than. Selects all fields with importance greater than the specified threshold.

Related information

- [Feature Selection node](#)
 - [Feature Selection Model Settings](#)
 - [Feature Selection Options](#)
 - [Feature Selection Model Nuggets](#)
 - [Feature Selection Model Results](#)
 - [Generating a Filter from a Feature Selection Model](#)
-

Generating a Filter from a Feature Selection Model

Based on the results of a Feature Selection model, you can use the Generate Filter from Feature dialog box to generate one or more Filter nodes that include or exclude subsets of fields based on importance relative to the specified target. While the model nugget can also be used as a filter, this gives you the flexibility to experiment with different subsets of fields without copying or modifying the model. The target field is always preserved by the filter regardless of whether include or exclude is selected.

Include/Exclude. You can choose to include or exclude fields—for example, to include the top 10 fields or exclude all fields marked as unimportant.

Selected fields. Includes or excludes all fields currently selected in the table.

All fields marked. Selects all fields marked as important, marginal, or unimportant.

Top number of fields. Allows you to select the top n fields based on importance.

Importance greater than. Selects all fields with importance greater than the specified threshold.

Related information

- [Feature Selection node](#)
 - [Feature Selection Model Settings](#)
 - [Feature Selection Options](#)
 - [Feature Selection Model Nuggets](#)
 - [Feature Selection Model Results](#)
 - [Selecting Fields by Importance](#)
-

Anomaly Detection Node

Anomaly detection models are used to identify outliers, or unusual cases, in the data. Unlike other modeling methods that store rules about unusual cases, anomaly detection models store information on what normal behavior looks like. This makes it possible to identify outliers even if they do not conform to any known pattern, and it can be particularly useful in applications, such as fraud detection, where new patterns may constantly be emerging. Anomaly detection is an unsupervised method, which means that it does not require a training dataset containing known cases of fraud to use as a starting point.

While traditional methods of identifying outliers generally look at one or two variables at a time, anomaly detection can examine large numbers of fields to identify clusters or peer groups into which similar records fall. Each record can then be compared to others in its peer group to identify possible anomalies. The further away a case is from the normal center, the more likely it is to be unusual. For example, the algorithm might lump records into three distinct clusters and flag those that fall far from the center of any one cluster.

Each record is assigned an anomaly index, which is the ratio of the group deviation index to its average over the cluster that the case belongs to. The larger the value of this index, the more deviation the case has than the average. Under the usual circumstance, cases with anomaly index values less than 1 or even 1.5 would not be considered as anomalies, because the deviation is just about the same or a bit more than the average. However, cases with an index value greater than 2 could be good anomaly candidates because the deviation is at least twice the average.

Anomaly detection is an exploratory method designed for quick detection of unusual cases or records that should be candidates for further analysis. These should be regarded as *suspected anomalies*, which, on closer examination, may or may not turn out to be real. You may find that a record is perfectly valid but choose to screen it from the data for purposes of model building. Alternatively, if the algorithm repeatedly turns up false anomalies, this may point to an error or artifact in the data collection process.

Note that anomaly detection identifies unusual records or cases through cluster analysis based on the set of fields selected in the model without regard for any specific target (dependent) field and regardless of whether those fields are relevant to the pattern you are trying to predict. For this reason, you may want to use anomaly detection in combination with feature selection or another technique for screening and ranking fields. For example, you can use feature selection to identify the most important fields relative to a specific target and then use anomaly detection to locate the records that are the most unusual with respect to those fields. (An alternative approach would be to build a decision tree model and then examine any misclassified records as potential anomalies. However, this method would be more difficult to replicate or automate on a large scale.)

Example. In screening agricultural development grants for possible cases of fraud, anomaly detection can be used to discover deviations from the norm, highlighting those records that are abnormal and worthy of further investigation. You are particularly interested in grant applications that seem to claim too much (or too little) money for the type and size of farm.

Requirements. One or more input fields. Note that only fields with a role set to Input using a source or Type node can be used as inputs. Target fields (role set to Target or Both) are ignored.

Strengths. By flagging cases that do *not* conform to a known set of rules rather than those that do, Anomaly Detection models can identify unusual cases even when they don't follow previously known patterns. When used in combination with feature selection, anomaly detection makes it possible to screen large amounts of data to identify the records of greatest interest relatively quickly.

- [Anomaly Detection Model Options](#)
- [Anomaly Detection Expert Options](#)

Anomaly Detection Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Determine cutoff value for anomaly based on. Specifies the method used to determine the cutoff value for flagging anomalies. The following options are available:

- **Minimum anomaly index level.** Specifies the minimum cutoff value for flagging anomalies. Records that meet or exceed this threshold are flagged.
- **Percentage of most anomalous records in the training data.** Automatically sets the threshold at a level that flags the specified percentage of records in the training data. The resulting cutoff is included as a parameter in the model. Note that this option determines how the cutoff value is set, *not* the actual percentage of records to be flagged during scoring. Actual scoring results may vary depending on the data.
- **Number of most anomalous records in the training data.** Automatically sets the threshold at a level that flags the specified number of records in the training data. The resulting threshold is included as a parameter in the model. Note that this option determines how the cutoff value is set, *not* the specific number of records to be flagged during scoring. Actual scoring results may vary depending on the data.

Note: Regardless of how the cutoff value is determined, it does not affect the underlying anomaly index value reported for each record. It simply specifies the threshold for flagging records as anomalous when estimating or scoring the model. If you later want to examine a larger or smaller number of records, you can use a Select node to identify a subset of records based on the anomaly index value (`$O-AnomalyIndex > x`).

Number of anomaly fields to report. Specifies the number of fields to report as an indication of why a particular record is flagged as an anomaly. The most anomalous fields are reported, defined as those that show the greatest deviation from the field norm for the cluster to which the record is assigned.

Anomaly Detection Expert Options

To specify options for missing values and other settings, set the mode to Expert on the Expert tab.

Adjustment coefficient. Value used to balance the relative weight given to continuous (numeric range) and categorical fields in calculating the distance. Larger values increase the influence of continuous fields. This must be a nonzero value.

Automatically calculate number of peer groups. Anomaly detection can be used to rapidly analyze a large number of possible solutions to choose the optimal number of peer groups for the training data. You can broaden or narrow the range by setting the minimum and maximum number of peer groups. Larger values will enable the system to explore a broader range of possible solutions; however, the cost is increased processing time.

Specify number of peer groups. If you know how many clusters to include in your model, select this option and enter the number of peer groups. Selecting this option will generally result in improved performance.

Noise level and ratio. These settings determine how outliers are treated during two-stage clustering. In the first stage, a cluster feature (CF) tree is used to condense the data from a very large number of individual records to a manageable number of clusters. The tree is built based on similarity measures, and when a node of the tree gets too many records in it, it splits into child nodes. In the second stage, hierarchical clustering commences on the terminal nodes of the CF tree. Noise handling is turned on in the first data pass, and it is off in the second data pass. The cases in the noise cluster from the first data pass are assigned to the regular clusters in the second data pass.

- **Noise level.** Specify a value between 0 and 0.5. This setting is relevant only if the CF tree fills during the growth phase, meaning that it cannot accept any more cases in a leaf node and that no leaf node can be split.
If the CF tree fills and the noise level is set to 0, the threshold will be increased and the CF tree regrown with all cases. After final clustering, values that cannot be assigned to a cluster are labeled outliers. The outlier cluster is given an identification number of -1. The outlier cluster is not included in the count of the number of clusters; that is, if you specify n clusters and noise handling, the algorithm will output n clusters and one noise cluster. In practical terms, increasing this value gives the algorithm more latitude to fit unusual records into the tree rather than assign them to a separate outlier cluster.

If the CF tree fills and the noise level is greater than 0, the CF tree will be regrown after placing any data in sparse leaves into their own noise leaf. A leaf is considered sparse if the ratio of the number of cases in the sparse leaf to the number of cases in the largest leaf is less than the noise level. After the tree is grown, the outliers will be placed in the CF tree if possible. If not, the outliers are discarded for the second phase of clustering.

- **Noise ratio.** Specifies the portion of memory allocated for the component that should be used for noise buffering. This value ranges between 0.0 and 0.5. If inserting a specific case into a leaf of the tree would yield tightness less than the threshold, the leaf is not split. If the tightness exceeds the threshold, the leaf is split, adding another small cluster to the CF tree. In practical terms, increasing this setting may cause the algorithm to gravitate more quickly toward a simpler tree.

Impute missing values. For continuous fields, substitutes the field mean in place of any missing values. For categorical fields, missing categories are combined and treated as a valid category. If this option is deselected, any records with missing values are excluded from the analysis.

Anomaly Detection Model Nuggets

Anomaly Detection model nuggets contain all of the information captured by the Anomaly Detection model as well as information about the training data and estimation process.

When you run a stream containing an Anomaly Detection model nugget, a number of new fields are added to the stream, as determined by the selections made on the Settings tab in the model nugget. See the topic [Anomaly Detection Model Settings](#) for more information. New field names are based on the model name, prefaced by $\$O$, as summarized in the following table.

Table 1. New field name generation

Field name	Description
$\$O\text{-Anomaly}$	Flag field indicating whether or not the record is anomalous.
$\$O\text{-AnomalyIndex}$	The anomaly index value for the record.
$\$O\text{-PeerGroup}$	Specifies the peer group to which the record is assigned.
$\$O\text{-Field-}n$	Name of the n th most anomalous field in terms of deviation from the cluster norm.
$\$O\text{-FieldImpact-}n$	Variable deviation index for the field. This value measures the deviation from the field norm for the cluster to which the record is assigned.

Optionally, you can suppress scores for non-anomalous records to make the results easier to read. See the topic [Anomaly Detection Model Settings](#) for more information.

- [Anomaly Detection model details](#)
- [Anomaly Detection Model Summary](#)
- [Anomaly Detection Model Settings](#)

Anomaly Detection model details

The Model tab for a generated Anomaly Detection model displays information about the peer groups in the model.

Note that the peer group sizes and statistics reported are estimates based on the training data and may differ slightly from actual scoring results even if run on the same data.

The Residual of the unreported reasons is 1 minus the sum of the mean anomaly index values for each of the anomaly columns for the records that are identified as anomalies. This percentage can serve as an indication of how much of the anomaly is explained by the fields that are

reported. This might guide you in determining how many anomaly fields to report.

Anomaly Detection Model Summary

The Summary tab for an Anomaly Detection model nugget displays information about the fields, build settings, and estimation process. The number of peer groups is also shown, along with the cutoff value used to flag records as anomalous.

Anomaly Detection Model Settings

Use the Settings tab to specify options for scoring the model nugget.

Indicate anomalous records with Specifies how anomalous records are treated in the output.

- Flag and index Creates a flag field that is set to *True* for all records that exceed the cutoff value included in the model. The anomaly index is also reported for each record in a separate field. See the topic [Anomaly Detection Model Options](#) for more information.
- Flag only Creates a flag field but without reporting the anomaly index for each record.
- Index only Reports the anomaly index without creating a flag field.

Number of anomaly fields to report Specifies the number of fields to report as an indication of why a particular record is flagged as an anomaly. The most anomalous fields are reported, defined as those that show the greatest deviation from the field norm for the cluster to which the record is assigned.

Discard records Select this option to discard all Non anomalous records from the stream, making it easier to focus on potential anomalies in any downstream nodes. Alternatively, you can choose to discard all Anomalous records in order to limit the subsequent analysis to those records that are not flagged as potential anomalies based on the model.

Note: Due to slight differences in rounding, the actual number of records flagged during scoring may not be identical to those flagged while training the model even if run on the same data.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Automated Modeling Nodes

The automated modeling nodes estimate and compare a number of different modeling methods, enabling you to try out a variety of approaches in a single modeling run. You can select the modeling algorithms to use, and the specific options for each, including combinations that would otherwise be mutually-exclusive. For example, rather than choose between the quick, dynamic, or prune methods for a Neural Net, you can try them all. The node explores every possible combination of options, ranks each candidate model based on the measure you specify, and saves the best for use in scoring or further analysis.

You can choose from three automated modeling nodes, depending on the needs of your analysis:

	The Auto Classifier node creates and compares a number of different models for binary outcomes (yes or no, churn or do not churn, and so on), allowing you to choose the best approach for a given analysis. A number of modeling algorithms are supported, making it possible to select the methods you want to use, the specific options for each, and the criteria for comparing the results. The node generates a set of models based on the specified options and ranks the best candidates according to the criteria you specify.
	The Auto Numeric node estimates and compares models for continuous numeric range outcomes using a number of different methods. The node works in the same manner as the Auto Classifier node, allowing you to choose the algorithms to use and to experiment with multiple combinations of options in a single modeling pass. Supported algorithms include neural networks, C&R Tree, CHAID, linear regression, generalized linear regression, and support vector machines (SVM). Models can be compared based on correlation, relative error, or number of variables used.
	The Auto Cluster node estimates and compares clustering models, which identify groups of records that have similar characteristics. The node works in the same manner as other automated modeling nodes, allowing you to experiment with multiple combinations of options in a single modeling pass. Models can be compared using basic measures with which to

attempt to filter and rank the usefulness of the cluster models, and provide a measure based on the importance of particular fields.

The best models are saved in a single composite model nugget, enabling you to browse and compare them, and to choose which models to use in scoring.

- For binary, nominal, and numeric targets only you can select multiple scoring models and combine the scores in a single model ensemble. By combining predictions from multiple models, limitations in individual models may be avoided, often resulting in a higher overall accuracy than can be gained from any one of the models.
- Optionally, you can choose to drill down into the results and generate modeling nodes or model nuggets for any of the individual models you want to use or explore further.

Models and Execution Time

Depending on the dataset and the number of models, automated modeling nodes may take hours or even longer to execute. When selecting options, pay attention to the number of models being produced. When practical, you may want to schedule modeling runs during nights or weekends when system resources are less likely to be in demand.

- If necessary, a Partition or Sample node can be used to reduce the number of records included in the initial training pass. Once you have narrowed the choices to a few candidate models, the full dataset can be restored.
 - To reduce the number of input fields, use Feature Selection. See the topic [Feature Selection node](#) for more information. Alternatively, you can use your initial modeling runs to identify fields and options that are worth exploring further. For example, if your best-performing models all seem to use the same three fields, this is a strong indication that those fields are worth keeping.
 - Optionally, you can limit the amount of time spent estimating any one model and specify the evaluation measures used to screen and rank models.
- [Automated Modeling Node Algorithm Settings](#)
 - [Automated Modeling Node Stopping Rules](#)
 - [Execution Feedback](#)
 - [Auto Classifier node](#)
 - [Auto Numeric node](#)
 - [Auto Cluster node](#)
 - [Automated Model Nuggets](#)

Automated Modeling Node Algorithm Settings

For each model type, you can use the default settings, or you can choose options for each model type. The specific options are similar to those available in the separate modeling nodes, with the difference that rather than choosing one setting or another, you can choose as many as you want to apply in most cases. For example, if comparing Neural Net models, you can choose several different training methods, and try each method with and without a random seed. All possible combinations of the selected options will be used, making it very easy to generate many different models in a single pass. Use care, however, as choosing multiple settings can cause the number of models to multiply very quickly.

To choose options for each model type

1. On the automated modeling node, select the Expert tab.
2. Click in the Model parameters column for the model type.
3. From the drop-down menu, choose Specify.
4. On the Algorithm settings dialog, select options from the Options column.

Note: Further options are available on the Expert tab of the Algorithm settings dialog.

Related information

- [Automated Modeling Node Stopping Rules](#)
- [Auto Classifier node](#)
- [Auto Classifier node model options](#)
- [Auto Classifier Node Expert Options](#)
- [Auto Classifier Node Discard Options](#)
- [Auto Classifier node settings](#)
- [Automated Model Nuggets](#)
- [Generating Nodes and Models](#)
- [Generating Evaluation Charts](#)
- [Evaluation Graphs](#)
- [Auto Numeric node](#)
- [Auto Numeric node model options](#)
- [Auto Numeric Node Expert Options](#)
- [Auto Numeric node settings](#)
- [Auto Cluster Node Model Options](#)

- [Auto Cluster Node Expert Options](#)
 - [Auto Cluster Node Discard Options](#)
-

Automated Modeling Node Stopping Rules

Stopping rules specified for automated modeling nodes relate to the overall node execution, not the stopping of individual models built by the node.

Restrict overall execution time. (Neural Network, K-Means, Kohonen, TwoStep, SVM, KNN, Bayes Net and C&R Tree models only) Stops execution after a specified number of hours. All models generated up to that point will be included in the model nugget, but no further models will be produced.

Stop as soon as valid models are produced. Stops execution when a model passes all criteria specified on the Discard tab (for the Auto Classifier or Auto Cluster node) or the Model tab (for the Auto Numeric node). See the topic [Auto Classifier Node Discard Options](#) for more information. See the topic [Auto Cluster Node Discard Options](#) for more information.

Related information

- [Automated Modeling Node Algorithm Settings](#)
 - [Auto Classifier node](#)
 - [Auto Classifier node model options](#)
 - [Auto Classifier Node Expert Options](#)
 - [Auto Classifier Node Discard Options](#)
 - [Auto Classifier node settings](#)
 - [Automated Model Nuggets](#)
 - [Generating Nodes and Models](#)
 - [Generating Evaluation Charts](#)
 - [Evaluation Graphs](#)
 - [Auto Numeric node](#)
 - [Auto Numeric node model options](#)
 - [Auto Numeric Node Expert Options](#)
 - [Auto Numeric node settings](#)
 - [Auto Cluster Node Model Options](#)
 - [Auto Cluster Node Expert Options](#)
 - [Auto Cluster Node Discard Options](#)
-

Execution Feedback

In addition to displaying the standard Execution Feedback dialog box if the stream takes longer than three seconds to execute, IBM® SPSS® Modeler displays information about the number of models involved.

Auto Classifier node

The Auto Classifier node estimates and compares models for either nominal (set) or binary (yes/no) targets, using a number of different methods, enabling you to try out a variety of approaches in a single modeling run. You can select the algorithms to use, and experiment with multiple combinations of options. For example, rather than choose between Radial Basis Function, polynomial, sigmoid, or linear methods for an SVM, you can try them all. The node explores every possible combination of options, ranks each candidate model based on the measure you specify, and saves the best models for use in scoring or further analysis. For more information, see [Automated Modeling Nodes](#).

Example

A retail company has historical data tracking the offers made to specific customers in past campaigns. The company now wants to achieve more profitable results by matching the right offer to each customer.

Requirements

A target field with a measurement level of either *Nominal* or *Flag* (with the role set to Target), and at least one input field (with the role set to Input). For a flag field, the *True* value defined for the target is assumed to represent a hit when calculating profits, lift, and related statistics. Input fields can have a measurement level of *Continuous* or *Categorical*, with the limitation that some inputs may not be appropriate for some model types. For example, ordinal fields used as inputs in C&R Tree, CHAID, and QUEST models must have numeric storage (not string), and will be ignored by these models if specified otherwise. Similarly, continuous input fields can be binned in some cases. The requirements are the same as when using the individual modeling nodes; for example a Bayes Net model works the same whether generated from the Bayes Net node or the Auto Classifier node.

Frequency and weight fields

Frequency and weight are used to give extra importance to some records over others because, for example, the user knows that the build dataset under-represents a section of the parent population (Weight) or because one record represents a number of identical cases (Frequency). If specified, a frequency field can be used by C&R Tree, CHAID, QUEST, Decision List, and Bayes Net models. A weight field can be used by C&RT, CHAID, and C5.0 models. Other model types will ignore these fields and build the models anyway. Frequency and weight fields are used only for model building, and are not considered when evaluating or scoring models. For more information, see [Using Frequency and Weight Fields](#).

Prefixes

If you attach a table node to the nugget for the Auto Classifier Node, there are several new variables in the table with names that begin with a \$ prefix.

The names of the fields that are generated during scoring are based on the target field, but with a standard prefix. Different model types use different sets of prefixes.

For example, the prefixes \$G, \$R, \$C are used as the prefix for predictions that are generated by the Generalized Linear model, CHAID model, and C5.0 model, respectively. \$X is typically generated by using an ensemble, and \$XR, \$XS, and \$XF are used as prefixes in cases where the target field is a Continuous, Categorical, or Flag field, respectively.

\$..C prefixes are used for prediction confidence of a Categorical, or Flag target; for example, \$XFC is used as a prefix for ensemble Flag prediction confidence. \$RC and \$CC are the prefixes for a single prediction of confidence for a CHAID model and C5.0 model respectively.

Supported Model Types

Supported model types include Neural Net, C&R Tree, QUEST, CHAID, C5.0, Logistic Regression, Decision List, Bayes Net, Discriminant, Nearest Neighbor, SVM, XGBoost Tree, and XGBoost-AS. See the topic [Auto Classifier Node Expert Options](#) for more information.

Continuous machine learning

An inconvenience with modeling is models getting outdated due to changes to your data over time. This is commonly referred to as *model drift* or *concept drift*. To help overcome model drift effectively, SPSS Modeler provides continuous automated machine learning. This feature is available for Auto Classifier node and Auto Numeric node model nuggets. For more information, see [Continuous machine learning](#).

- [Auto Classifier node model options](#)
 - [Auto Classifier Node Expert Options](#)
 - [Auto Classifier Node Discard Options](#)
 - [Auto Classifier node settings](#)
-

Auto Classifier node model options

The Model tab of the Auto Classifier node enables you to specify the number of models to be created, along with the criteria used to compare models.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Cross-validate. Cross-validation gives the model a dataset of *known data* on which to run training (a training dataset), and a dataset of *unknown data* to test the model against (validation dataset or testing set). The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias.

Create split models. Builds a separate model for each possible value of input fields that are specified as split fields. See [Building Split Models](#) for more information.

Rank models by. Specifies the criteria used to compare and rank models. Options include overall accuracy, area under the ROC curve, profit, lift, and number of fields. Note that all of these measures will be available in the summary report regardless of which is selected here.

Note: For a nominal (set) target, ranking is restricted to either Overall Accuracy or Number of Fields.

When calculating profits, lift, and related statistics, the *True* value defined for the target field is assumed to represent a hit.

- Overall accuracy. The percentage of records that is correctly predicted by the model relative to the total number of records.
- Area under the ROC curve. The ROC curve provides an index for the performance of a model. The further the curve lies above the reference line, the more accurate the test.
- Profit (Cumulative). The sum of profits across cumulative percentiles (sorted in terms of confidence for the prediction), as computed based on the specified cost, revenue, and weight criteria. Typically, the profit starts near 0 for the top percentile, increases steadily, and then decreases. For a good model, profits will show a well-defined peak, which is reported along with the percentile where it occurs. For a model that provides no information, the profit curve will be relatively straight and may be increasing, decreasing, or level, depending on the cost/revenue structure that applies.

- Lift (Cumulative). The ratio of hits in cumulative quantiles relative to the overall sample (where quantiles are sorted in terms of confidence for the prediction). For example, a lift value of 3 for the top quantile indicates a hit rate three times as high as for the sample overall. For a good model, lift should start well above 1.0 for the top quantiles and then drop off sharply toward 1.0 for the lower quantiles. For a model that provides no information, the lift will hover around 1.0.
- Number of fields. Ranks models based on the number of input fields used.

Rank models using. If a partition is in use, you can specify whether ranks are based on the training dataset or the testing set. With large datasets, use of a partition for preliminary screening of models may greatly improve performance.

Number of models to use. Specifies the maximum number of models to be listed in the model nugget produced by the node. The top-ranking models are listed according to the specified ranking criterion. Note that increasing this limit may slow performance. The maximum allowable value is 100.

Calculate predictor importance. For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may extend the time needed to calculate some models, and is not recommended if you simply want a broad comparison across many different models. It is more useful once you have narrowed your analysis to a handful of models that you want to explore in greater detail. See [Predictor Importance](#) for more information.

Profit Criteria. Only for flag targets. Profit equals the revenue for each record minus the cost for the record. Profits for a quantile are simply the sum of profits for all records in the quantile. Profits are assumed to apply only to hits, but costs apply to all records.

- Costs. Specify the cost associated with each record. You can select Fixed or Variable costs. For fixed costs, specify the cost value. For variable costs, click the Field Chooser button to select a field as the cost field. (Costs is not available for ROC charts.)
- Revenue. Specify the revenue associated with each record that represents a hit. You can select Fixed or Variable costs. For fixed revenue, specify the revenue value. For variable revenue, click the Field Chooser button to select a field as the revenue field. (Revenue is not available for ROC charts.)
- Weight. If the records in your data represent more than one unit, you can use frequency weights to adjust the results. Specify the weight associated with each record, using Fixed or Variable weights. For fixed weights, specify the weight value (the number of units per record). For variable weights, click the Field Chooser button to select a field as the weight field. (Weight is not available for ROC charts.)

Lift Criteria. Only for flag targets. Specifies the percentile to use for lift calculations. Note that you can also change this value when comparing the results. See the topic [Automated Model Nuggets](#) for more information.

Auto Classifier Node Expert Options

The Expert tab of the Auto Classifier node enables you to apply a partition (if available), select the algorithms to use, and specify stopping rules.

Select models. By default, all models are selected to be built; however, if you have Analytic Server, you can choose to restrict the models to those that can run on Analytic Server and preset them so that they either build split models or are ready to process very large data sets.

Note: Local building of Analytic Server models within the Auto Classifier node is not supported.

Models used. Use the check boxes in the column on the left to select the model types (algorithms) to include in the comparison. The more types you select, the more models will be created and the longer the processing time will be.

Model type. Lists the available algorithms (see below).

Model parameters. For each model type, you can use the default settings or select Specify to choose options for each model type. The specific options are similar to those available in the separate modeling nodes, with the difference that multiple options or combinations can be selected. For example, if comparing Neural Net models, rather than choosing one of the six training methods, you can choose all of them to train six models in a single pass.

Number of models. Lists the number of models produced for each algorithm based on current settings. When combining options, the number of models can quickly add up, so paying close attention to this number is strongly recommended, particularly when using large datasets.

Restrict maximum time spent building a single model. (K-Means, Kohonen, TwoStep, SVM, KNN, Bayes Net and Decision List models only) Sets a maximum time limit for any one model. For example, if a particular model requires an unexpectedly long time to train because of some complex interaction, you probably don't want it to hold up your entire modeling run.

Note: If the target is a nominal (set) field, the Decision List option is unavailable.

Supported Algorithms

	The Support Vector Machine (SVM) node enables you to classify data into one of two groups without overfitting. SVM works well with wide data sets, such as those with a very large number of input fields.
	The <i>k</i> -Nearest Neighbor (KNN) node associates a new case with the category or value of the <i>k</i> objects nearest to it in the predictor space, where <i>k</i> is an integer. Similar cases are near each other and dissimilar cases are distant from each other.

	Discriminant analysis makes more stringent assumptions than logistic regression but can be a valuable alternative or supplement to a logistic regression analysis when those assumptions are met.
	The Bayesian Network node enables you to build a probability model by combining observed and recorded evidence with real-world knowledge to establish the likelihood of occurrences. The node focuses on Tree Augmented Naïve Bayes (TAN) and Markov Blanket networks that are primarily used for classification.
	The Decision List node identifies subgroups, or segments, that show a higher or lower likelihood of a given binary outcome relative to the overall population. For example, you might look for customers who are unlikely to churn or are most likely to respond favorably to a campaign. You can incorporate your business knowledge into the model by adding your own custom segments and previewing alternative models side by side to compare the results. Decision List models consist of a list of rules in which each rule has a condition and an outcome. Rules are applied in order, and the first rule that matches determines the outcome.
	Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric range.
	The CHAID node generates decision trees using chi-square statistics to identify optimal splits. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.
	The QUEST node provides a binary classification method for building decision trees, designed to reduce the processing time required for large C&R Tree analyses while also reducing the tendency found in classification tree methods to favor inputs that allow more splits. Input fields can be numeric ranges (continuous), but the target field must be categorical. All splits are binary.
	The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered “pure” if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).
	The C5.0 node builds either a decision tree or a rule set. The model works by splitting the sample based on the field that provides the maximum information gain at each level. The target field must be categorical. Multiple splits into more than two subgroups are allowed.
	The Neural Net node uses a simplified model of the way the human brain processes information. It works by simulating a large number of interconnected simple processing units that resemble abstract versions of neurons. Neural networks are powerful general function estimators and require minimal statistical or mathematical knowledge to train or apply.
	Linear regression models predict a continuous target based on linear relationships between the target and one or more predictors.
	The Linear Support Vector Machine (LSVM) node enables you to classify data into one of two groups without overfitting. LSVM is linear and works well with wide data sets, such as those with a very large number of records.
	The Random Trees node is similar to the existing C&RT node; however, the Random Trees node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS® Modeler version 17. The Random Trees tree node generates a decision tree that you use to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered <i>pure</i> if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).
	The Tree-AS node is similar to the existing CHAID node; however, the Tree-AS node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS Modeler version 17. The node generates a decision tree by using chi-square statistics (CHAID) to identify optimal splits. This use of CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.
	XGBoost Tree® is an advanced implementation of a gradient boosting algorithm with a tree model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost Tree is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost Tree node in SPSS Modeler exposes the core features and commonly used parameters. The node is implemented in Python.
	XGBoost® is an advanced implementation of a gradient boosting algorithm. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost-AS node in SPSS Modeler exposes the core features and commonly used parameters. The XGBoost-AS node is implemented in Spark.

Note: If you select Tree-AS to run on Analytic Server, it will fail to build a model when there is a Partition node upstream. In this case, to make Auto Classifier work with other modeling nodes on Analytic Server, deselect the Tree-AS model type.

Auto Classifier Node Discard Options

The Discard tab of the Auto Classifier node enables you to automatically discard models that do not meet certain criteria. These models will not be listed in the summary report.

You can specify a minimum threshold for overall accuracy and a maximum threshold for the number of variables used in the model. In addition, for flag targets, you can specify a minimum threshold for lift, profit, and area under the curve; lift and profit are determined as specified on the Model tab. See the topic [Auto Classifier node model options](#) for more information.

Optionally, you can configure the node to stop execution the first time a model is generated that meets all specified criteria. See the topic [Automated Modeling Node Stopping Rules](#) for more information.

Related information

- [Automated Modeling Node Algorithm Settings](#)
 - [Automated Modeling Node Stopping Rules](#)
 - [Auto Classifier node](#)
 - [Auto Classifier node model options](#)
 - [Auto Classifier Node Expert Options](#)
 - [Auto Classifier node settings](#)
 - [Automated Model Nuggets](#)
 - [Generating Nodes and Models](#)
 - [Generating Evaluation Charts](#)
 - [Evaluation Graphs](#)
-

Auto Classifier node settings

On the Auto Classifier node's Settings tab, you can preconfigure the score-time options that are available on the nugget.

Filter out fields generated by ensembled models. Removes from the output all of the additional fields generated by the individual models that feed into the Ensemble node. Select this check box if you are interested only in the combined score from all of the input models. Ensure that this option is deselected if, for example, you want to use an Analysis node or Evaluation node to compare the accuracy of the combined score with that of each of the individual input models.

Tip: *Continuous machine learning* settings are also available for the Auto Classifier node and the Auto Numeric node. For details, see [Continuous machine learning](#).

Auto Numeric node

The Auto Numeric node estimates and compares models for continuous numeric range outcomes using a number of different methods, enabling you to try out a variety of approaches in a single modeling run. You can select the algorithms to use, and experiment with multiple combinations of options. For example, you could predict housing values using neural net, linear regression, C&RT, and CHAID models to see which performs best, and you could try out different combinations of stepwise, forward, and backward regression methods. The node explores every possible combination of options, ranks each candidate model based on the measure you specify, and saves the best for use in scoring or further analysis. See the topic [Automated Modeling Nodes](#) for more information.

Example

A municipality wants to more accurately estimate real estate taxes and to adjust values for specific properties as needed without having to inspect every property. Using the Auto Numeric node, the analyst can generate and compare a number of models that predict property values based on building type, neighborhood, size, and other known factors.

Requirements

A single target field (with the role set to Target), and at least one input field (with the role set to Input). The target must be a continuous (numeric range) field, such as *age* or *income*. Input fields can be continuous or categorical, with the limitation that some inputs may not be appropriate for some model types. For example, C&R Tree models can use categorical string fields as inputs, while linear regression models cannot use these fields and will ignore them if specified. The requirements are the same as when using the individual modeling nodes. For example, a CHAID model works the same whether generated from the CHAID node or the Auto Numeric node.

Frequency and weight fields

Frequency and weight are used to give extra importance to some records over others because, for example, the user knows that the build dataset under-represents a section of the parent population (Weight) or because one record represents a number of identical cases (Frequency). If specified, a frequency field can be used by C&R Tree and CHAID algorithms. A weight field can be used by C&RT, CHAID, Regression, and GenLin algorithms. Other model types will ignore these fields and build the models anyway. Frequency and weight fields are used only for model building and are not considered when evaluating or scoring models. See the topic [Using Frequency and Weight Fields](#) for more information.

Prefixes

If you attach a table node to the nugget for the Auto Numeric Node, there are several new variables in the table with names that begin with a \$ prefix.

The names of the fields that are generated during scoring are based on the target field, but with a standard prefix. Different model types use different sets of prefixes.

For example, the prefixes \$G, \$R, \$C are used as the prefix for predictions that are generated by the Generalized Linear model, CHAID model, and C5.0 model, respectively. \$X is typically generated by using an ensemble, and \$XR, \$XS, and \$XF are used as prefixes in cases where the target field is a Continuous, Categorical, or Flag field, respectively.

\$..E prefixes are used for the prediction confidence of a Continuous target; for example, \$XRE is used as a prefix for ensemble Continuous prediction confidence. \$GE is the prefix for a single prediction of confidence for a Generalized Linear model.

Supported model types

Supported model types include Neural Net, C&R Tree, CHAID, Regression, GenLin, Nearest Neighbor, SVM, XGBoost Linear, GLE, and XGBoost-AS. For more information, see [Auto Numeric Node Expert Options](#).

Continuous machine learning

An inconvenience with modeling is models getting outdated due to changes to your data over time. This is commonly referred to as *model drift* or *concept drift*. To help overcome model drift effectively, SPSS Modeler provides continuous automated machine learning. This feature is available for Auto Classifier node and Auto Numeric node model nuggets. For more information, see [Continuous machine learning](#).

- [Auto Numeric node model options](#)
 - [Auto Numeric Node Expert Options](#)
 - [Auto Numeric node settings](#)
-

Auto Numeric node model options

The Model tab of the Auto Numeric node enables you to specify the number of models to be saved, along with the criteria used to compare models.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Cross-validate. Cross-validation gives the model a dataset of *known data* on which to run training (a training dataset), and a dataset of *unknown data* to test the model against (validation dataset or testing set). The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias.

Create split models. Builds a separate model for each possible value of input fields that are specified as split fields. See [Building Split Models](#) for more information.

Rank models by. Specifies the criteria used to compare models.

- Correlation. The Pearson Correlation between the observed value for each record and the value predicted by the model. The correlation is a measure of linear association between two variables, with values closer to 1 indicating a stronger relationship. (Correlation values range between -1, for a perfect negative relationship, and +1 for a perfect positive relationship. A value of 0 indicates no linear relationship, while a model with a negative correlation would rank lowest of all.)
- Number of fields. The number of fields used as predictors in the model. Choosing models that use fewer fields may streamline data preparation and improve performance in some cases.
- Relative error. The relative error is the ratio of the variance of the observed values from those predicted by the model to the variance of the observed values from the mean. In practical terms, it compares how well the model performs relative to a **null** or **intercept** model that simply returns the mean value of the target field as the prediction. For a good model, this value should be less than 1, indicating that the model is more accurate than the null model. A model with a relative error greater than 1 is less accurate than the null model and is therefore not useful. For linear regression models, the relative error is equal to the square of the correlation and adds no new information. For nonlinear models, the relative error is unrelated to the correlation and provides an additional measure for assessing model performance.

Rank models using. If a partition is in use, you can specify whether ranks are based on the training partition or the testing partition. With large datasets, use of a partition for preliminary screening of models may greatly improve performance.

Number of models to use. Specifies the maximum number of models to be shown in the model nugget produced by the node. The top-ranking models are listed according to the specified ranking criterion. Increasing this limit will enable you to compare results for more models but may slow performance. The maximum allowable value is 100.

Calculate predictor importance. For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may extend the time needed to calculate some

models, and is not recommended if you simply want a broad comparison across many different models. It is more useful once you have narrowed your analysis to a handful of models that you want to explore in greater detail. See [Predictor Importance](#) for more information.

Do not keep models if. Specifies threshold values for correlation, relative error, and number of fields used. Models that fail to meet any of these criteria will be discarded and will not be listed in the summary report.

- Correlation less than. The minimum correlation (in terms of absolute value) for a model to be included in the summary report.
- Number of fields used is greater than. The maximum number of fields to be used by any model to be included.
- Relative error is greater than. The maximum relative error for any model to be included.

Optionally, you can configure the node to stop execution the first time a model is generated that meets all specified criteria. See the topic [Automated Modeling Node Stopping Rules](#) for more information.

Auto Numeric Node Expert Options

The Expert tab of the Auto Numeric node enables you to select the algorithms and options to use and to specify stopping rules.

Select models. By default, all models are selected to be built; however, if you have Analytic Server, you can choose to restrict the models to those that can run on Analytic Server and preset them so that they either build split models or are ready to process very large data sets.

Note: Local building of Analytic Server models within the Auto Numeric node is not supported.

Models used. Use the check boxes in the column on the left to select the model types (algorithms) to include in the comparison. The more types you select, the more models will be created and the longer the processing time will be.

Model type. Lists the available algorithms (see below).

Model parameters. For each model type, you can use the default settings or select Specify to choose options for each model type. The specific options are similar to those available in the separate modeling nodes, with the difference that multiple options or combinations can be selected. For example, if comparing Neural Net models, rather than choosing one of the six training methods, you can choose all of them to train six models in a single pass.

Number of models. Lists the number of models produced for each algorithm based on current settings. When combining options, the number of models can quickly add up, so paying close attention to this number is strongly recommended, particularly when using large datasets.

Restrict maximum time spent building a single model. (K-Means, Kohonen, TwoStep, SVM, KNN, Bayes Net and Decision List models only) Sets a maximum time limit for any one model. For example, if a particular model requires an unexpectedly long time to train because of some complex interaction, you probably don't want it to hold up your entire modeling run.

Supported algorithms

	The Neural Net node uses a simplified model of the way the human brain processes information. It works by simulating a large number of interconnected simple processing units that resemble abstract versions of neurons. Neural networks are powerful general function estimators and require minimal statistical or mathematical knowledge to train or apply.
	The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered "pure" if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).
	The CHAID node generates decision trees using chi-square statistics to identify optimal splits. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.
	Linear regression is a common statistical technique for summarizing data and making predictions by fitting a straight line or surface that minimizes the discrepancies between predicted and actual output values.
	The Generalized Linear model expands the general linear model so that the dependent variable is linearly related to the factors and covariates through a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers the functionality of a wide number of statistical models, including linear regression, logistic regression, loglinear models for count data, and interval-censored survival models.
	The k -Nearest Neighbor (KNN) node associates a new case with the category or value of the k objects nearest to it in the predictor space, where k is an integer. Similar cases are near each other and dissimilar cases are distant from each other.
	The Support Vector Machine (SVM) node enables you to classify data into one of two groups without overfitting. SVM works well with wide data sets, such as those with a very large number of input fields.
	Linear regression models predict a continuous target based on linear relationships between the target and one or more predictors.

	The Linear Support Vector Machine (LSVM) node enables you to classify data into one of two groups without overfitting. LSVM is linear and works well with wide data sets, such as those with a very large number of records.
	The Random Trees node is similar to the existing C&RT node; however, the Random Trees node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS® Modeler version 17. The Random Trees tree node generates a decision tree that you use to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered <i>pure</i> if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).
	The Tree-AS node is similar to the existing CHAID node; however, the Tree-AS node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS Modeler version 17. The node generates a decision tree by using chi-square statistics (CHAID) to identify optimal splits. This use of CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.
	XGBoost Linear® is an advanced implementation of a gradient boosting algorithm with a linear model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. The XGBoost Linear node in SPSS Modeler is implemented in Python.
	A GLE extends the linear model so that the target can have a non-normal distribution, is linearly related to the factors and covariates via a specified link function, and so that the observations can be correlated. Generalized linear mixed models cover a wide variety of models, from simple linear regression to complex multilevel models for non-normal longitudinal data.
	XGBoost® is an advanced implementation of a gradient boosting algorithm. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost-AS node in SPSS Modeler exposes the core features and commonly used parameters. The XGBoost-AS node is implemented in Spark.

Auto Numeric node settings

The Settings tab of the Auto Numeric node enables you to preconfigure the score-time options that are available on the nugget.

Filter out fields generated by ensembled models. Removes from the output all of the additional fields generated by the individual models that feed into the Ensemble node. Select this check box if you are interested only in the combined score from all of the input models. Ensure that this option is deselected if, for example, you want to use an Analysis node or Evaluation node to compare the accuracy of the combined score with that of each of the individual input models.

Calculate standard error. For a continuous (numeric range) target, a standard error calculation is run by default to calculate the difference between the measured or estimated values and the true values; and to show how close those estimates matched.

Tip: *Continuous machine learning* settings are also available for the Auto Classifier node and the Auto Numeric node. For details, see [Continuous machine learning](#).

Auto Cluster node

The Auto Cluster node estimates and compares clustering models that identify groups of records with similar characteristics. The node works in the same manner as other automated modeling nodes, enabling you to experiment with multiple combinations of options in a single modeling pass. Models can be compared using basic measures with which to attempt to filter and rank the usefulness of the cluster models, and provide a measure based on the importance of particular fields.

Clustering models are often used to identify groups that can be used as inputs in subsequent analyses. For example you may want to target groups of customers based on demographic characteristics such as income, or based on the services they have bought in the past. This can be done without prior knowledge about the groups and their characteristics -- you may not know how many groups to look for, or what features to use in defining them. Clustering models are often referred to as unsupervised learning models, since they do not use a target field, and do not return a specific prediction that can be evaluated as true or false. The value of a clustering model is determined by its ability to capture interesting groupings in the data and provide useful descriptions of those groupings. See [Clustering models](#) for more information.

Requirements. One or more fields that define characteristics of interest. Cluster models do not use target fields in the same manner as other models, because they do not make specific predictions that can be assessed as true or false. Instead they are used to identify groups of cases that may be related. For example you cannot use a cluster model to predict whether a given customer will churn or respond to an offer. But you can use a cluster model to assign customers to groups based on their tendency to do those things. Weight and frequency fields are not used.

Evaluation fields. While no target is used, you can optionally specify one or more evaluation fields to be used in comparing models. The usefulness of a cluster model may be evaluated by measuring how well (or badly) the clusters differentiate these fields.

Supported model types

Supported model types include TwoStep, K-Means, Kohonen, One-Class SVM, and K-Means-AS.

- [Auto Cluster Node Model Options](#)
 - [Auto Cluster Node Expert Options](#)
 - [Auto Cluster Node Discard Options](#)
-

Auto Cluster Node Model Options

The Model tab of the Auto Cluster node enables you to specify the number of models to be saved, along with the criteria used to compare models.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Rank models by. Specifies the criteria used to compare and rank models.

- **Silhouette.** An index measuring both cluster cohesion and separation. See *Silhouette Ranking Measure* below for more information.
- **Number of clusters.** The number of clusters in the model.
- **Size of smallest cluster.** The smallest cluster size.
- **Size of largest cluster.** The largest cluster size.
- **Smallest / largest cluster.** The ratio of the size of the smallest cluster to the largest cluster.
- **Importance.** The importance of the Evaluation field on the Fields tab. Note that this can only be calculated if an Evaluation field has been specified.

Rank models using. If a partition is in use, you can specify whether ranks are based on the training dataset or the testing set. With large datasets, use of a partition for preliminary screening of models may greatly improve performance.

Number of models to keep. Specifies the maximum number of models to be listed in the nugget produced by the node. The top-ranking models are listed according to the specified ranking criterion. Note that increasing this limit may slow performance. The maximum allowable value is 100.

Silhouette Ranking Measure

The default ranking measure, Silhouette, has a default value of 0 because a value of less than 0 (i.e. negative) indicates that the average distance between a case and points in its assigned cluster is greater than the minimum average distance to points in another cluster. Therefore, models with a negative Silhouette can safely be discarded.

The ranking measure is actually a modified silhouette coefficient, which combines the concepts of cluster cohesion (favoring models which contain tightly cohesive clusters) and cluster separation (favoring models which contain highly separated clusters). The average Silhouette coefficient is simply the average over all cases of the following calculation for each individual case:

$$(B - A) / \max(A, B)$$

where A is the distance from the case to the centroid of the cluster which the case belongs to; and B is the minimal distance from the case to the centroid of every other cluster.

The Silhouette coefficient (and its average) range between -1 (indicating a very poor model) and 1 (indicating an excellent model). The average can be conducted on the level of total cases (which produces total Silhouette) or the level of clusters (which produces cluster Silhouette). Distances may be calculated using Euclidean distances.

Related information

- [Automated Modeling Node Algorithm Settings](#)
 - [Automated Modeling Node Stopping Rules](#)
 - [Auto Cluster Node Expert Options](#)
 - [Auto Cluster Node Discard Options](#)
 - [Automated Model Nuggets](#)
 - [Generating Nodes and Models](#)
 - [Generating Evaluation Charts](#)
-

Auto Cluster Node Expert Options

The Expert tab of the Auto Cluster node enables you to apply a partition (if available), select the algorithms to use, and specify stopping rules.

Select models. By default, all models are selected to be built; however, if you have Analytic Server, you can choose to restrict the models to those that can run on Analytic Server and preset them so that they either build split models or are ready to process very large data sets.

Note: Local building of Analytic Server models within the Auto Cluster node is not supported.

Models used. Use the check boxes in the column on the left to select the model types (algorithms) to include in the comparison. The more types you select, the more models will be created and the longer the processing time will be.

Model type. Lists the available algorithms (see below).

Model parameters. For each model type, you can use the default settings or select Specify to choose options for each model type. The specific options are similar to those available in the separate modeling nodes, with the difference that multiple options or combinations can be selected. For example, if comparing Neural Net models, rather than choosing one of the six training methods, you can choose all of them to train six models in a single pass.

Number of models. Lists the number of models produced for each algorithm based on current settings. When combining options, the number of models can quickly add up, so paying close attention to this number is strongly recommended, particularly when using large datasets.

Restrict maximum time spent building a single model. (K-Means, Kohonen, TwoStep, SVM, KNN, Bayes Net and Decision List models only) Sets a maximum time limit for any one model. For example, if a particular model requires an unexpectedly long time to train because of some complex interaction, you probably don't want it to hold up your entire modeling run.

Supported algorithms

	The K-Means node clusters the data set into distinct groups (or clusters). The method defines a fixed number of clusters, iteratively assigns records to clusters, and adjusts the cluster centers until further refinement can no longer improve the model. Instead of trying to predict an outcome, k-means uses a process known as unsupervised learning to uncover patterns in the set of input fields.
	The Kohonen node generates a type of neural network that can be used to cluster the data set into distinct groups. When the network is fully trained, records that are similar should be close together on the output map, while records that are different will be far apart. You can look at the number of observations captured by each unit in the model nugget to identify the strong units. This may give you a sense of the appropriate number of clusters.
	The TwoStep node uses a two-step clustering method. The first step makes a single pass through the data to compress the raw input data into a manageable set of subclusters. The second step uses a hierarchical clustering method to progressively merge the subclusters into larger and larger clusters. TwoStep has the advantage of automatically estimating the optimal number of clusters for the training data. It can handle mixed field types and large data sets efficiently.

Auto Cluster Node Discard Options

The Discard tab of the Auto Cluster node enables you to automatically discard models that do not meet certain criteria. These models will not be listed on the model nugget.

You can specify the minimum silhouette value, cluster numbers, cluster sizes, and the importance of the evaluation field used in the model. Silhouette and the number and size of clusters are determined as specified in the modeling node. See the topic [Auto Cluster Node Model Options](#) for more information.

Optionally, you can configure the node to stop execution the first time a model is generated that meets all specified criteria. See the topic [Automated Modeling Node Stopping Rules](#) for more information.

Related information

- [Automated Modeling Node Algorithm Settings](#)
- [Automated Modeling Node Stopping Rules](#)
- [Auto Cluster Node Model Options](#)
- [Auto Cluster Node Expert Options](#)
- [Automated Model Nuggets](#)
- [Generating Nodes and Models](#)
- [Generating Evaluation Charts](#)

Automated Model Nuggets

When an automated modeling node is executed, the node estimates candidate models for every possible combination of options, ranks each candidate model based on the measure you specify, and saves the best models in a composite automated model nugget. This model nugget

actually contains a set of one or more models generated by the node, which can be individually browsed or selected for use in scoring. The model type and build time are listed for each model, along with a number of other measures as appropriate for the type of model. You can sort the table on any of these columns to quickly identify the most interesting models.

- To browse any of the individual model nuggets, double-click the nugget icon. From there you can then generate a modeling node for that model to the stream canvas, or a copy of the model nugget to the models palette.
- Thumbnail graphs give a quick visual assessment for each model type, as summarized below. You can double-click on a thumbnail to generate a full-sized graph. The full-sized plot shows up to 1000 points and will be based on a sample if the dataset contains more. (For scatterplots only, the graph is regenerated each time it is displayed, so any changes in the upstream data—such as updating of a random sample or partition if Set Random Seed is not selected—may be reflected each time the scatterplot is redrawn.)
- Use the toolbar to show or hide specific columns on the Model tab or to change the column used to sort the table. (You can also change the sort by clicking on the column headers.)
- Use the Delete button to permanently remove any unused models.
- To reorder columns, click on a column header and drag the column to the desired location.
- If a partition is in use, you can choose to view results for the training or testing partition as applicable.

The specific columns depend on the type of models being compared, as detailed below.

Binary Targets

- For binary models, the thumbnail graph shows the distribution of actual values, overlaid with the predicted values, to give a quick visual indication of how many records were correctly predicted in each category.
- Ranking criteria match the options in the Auto Classifier modeling node. See the topic [Auto Classifier node model options](#) for more information.
- For the maximum profit, the percentile in which the maximum occurs is also reported.
- For cumulative lift, you can change the selected percentile using the toolbar.

Nominal Targets

- For nominal (set) models, the thumbnail graph shows the distribution of actual values, overlaid with the predicted values, to give a quick visual indication of how many records were correctly predicted in each category.
- Ranking criteria match the options in the Auto Classifier modeling node. See the topic [Auto Classifier node model options](#) for more information.

Continuous Targets

- For continuous (numeric range) models, the graph plots predicted against observed values for each model, providing a quick visual indication of the correlation between them. For a good model, points should tend to cluster along the diagonal rather than be scattered randomly across the graph.
- Ranking criteria match the options in the Auto Numeric modeling node. See the topic [Auto Numeric node model options](#) for more information.

Cluster Targets

- For cluster models, the graph plots counts against clusters for each model, providing a quick visual indication of cluster distribution.
- Ranking criteria match the options in the Auto Cluster modeling node. See the topic [Auto Cluster Node Model Options](#) for more information.

Selecting Models for Scoring

The Use? column enables you to select the models to use in scoring.

- For binary, nominal, and numeric targets, you can select multiple scoring models and combine the scores in the single, ensembled model nugget. By combining predictions from multiple models, limitations in individual models may be avoided, often resulting in a higher overall accuracy than can be gained from any one of the models.
 - For cluster models, only one scoring model can be selected at a time. By default, the top ranked one is selected first.
- [Generating Nodes and Models](#)
 - [Generating Evaluation Charts](#)
 - [Evaluation Graphs](#)
 - [Automated Model Nugget Summary](#)
 - [Continuous machine learning](#)

As a result of IBM research, and inspired by natural selection in biology, *continuous machine learning* is available for the Auto Classifier node and the Auto Numeric node.

Related information

- [Automated Modeling Node Algorithm Settings](#)
- [Automated Modeling Node Stopping Rules](#)
- [Auto Classifier node](#)
- [Auto Classifier node model options](#)

- [Auto Classifier Node Expert Options](#)
 - [Auto Classifier Node Discard Options](#)
 - [Auto Classifier node settings](#)
 - [Generating Nodes and Models](#)
 - [Generating Evaluation Charts](#)
 - [Evaluation Graphs](#)
 - [Auto Numeric node](#)
 - [Auto Numeric node model options](#)
 - [Auto Numeric Node Expert Options](#)
 - [Auto Numeric node settings](#)
 - [Auto Cluster Node Model Options](#)
 - [Auto Cluster Node Expert Options](#)
 - [Auto Cluster Node Discard Options](#)
-

Generating Nodes and Models

You can generate a copy of the composite automated model nugget, or the automated modeling node from which it was built. For example, this may be useful if you do not have the original stream from which the automated model nugget was built. Alternatively, you can generate a nugget or modeling node for any of the individual models listed in the automated model nugget.

Automated Modeling Nugget

From the Generate menu, select Model to Palette to add the automated model nugget to the Models palette. The generated model can be saved or used as is without rerunning the stream.

Alternatively, you can select Generate Modeling Node from the Generate menu to add the modeling node to the stream canvas. This node can be used to reestimate the selected models without repeating the entire modeling run.

Individual Modeling Nugget

1. In the Model menu, double-click on the individual nugget you require. A copy of that nugget opens in a new dialog.
2. From the Generate menu in the new dialog, select Model to Palette to add the individual modeling nugget to the Models palette.
3. Alternatively, you can select Generate Modeling Node from the Generate menu in the new dialog to add the individual modeling node to the stream canvas.

Related information

- [Automated Modeling Node Algorithm Settings](#)
 - [Automated Modeling Node Stopping Rules](#)
 - [Auto Classifier node](#)
 - [Auto Classifier node model options](#)
 - [Auto Classifier Node Expert Options](#)
 - [Auto Classifier Node Discard Options](#)
 - [Auto Classifier node settings](#)
 - [Automated Model Nuggets](#)
 - [Generating Evaluation Charts](#)
 - [Evaluation Graphs](#)
 - [Auto Numeric node](#)
 - [Auto Numeric node model options](#)
 - [Auto Numeric Node Expert Options](#)
 - [Auto Numeric node settings](#)
 - [Auto Cluster Node Model Options](#)
 - [Auto Cluster Node Expert Options](#)
 - [Auto Cluster Node Discard Options](#)
-

Generating Evaluation Charts

For binary models only, you can generate evaluation charts that offer a visual way to assess and compare the performance of each model. Evaluation charts are not available for models generated by the Auto Numeric or Auto Cluster nodes.

1. Under the Use? column in the Auto Classifier automated model nugget, select the models that you want to evaluate.
2. From the Generate menu, choose Evaluation Chart(s). The Evaluation Chart dialog box is displayed.
3. Select the chart type and other options as desired.

Related information

- [Automated Modeling Node Algorithm Settings](#)
 - [Automated Modeling Node Stopping Rules](#)
 - [Auto Classifier node](#)
 - [Auto Classifier node model options](#)
 - [Auto Classifier Node Expert Options](#)
 - [Auto Classifier Node Discard Options](#)
 - [Auto Classifier node settings](#)
 - [Automated Model Nuggets](#)
 - [Generating Nodes and Models](#)
 - [Evaluation Graphs](#)
 - [Auto Numeric node](#)
 - [Auto Numeric node model options](#)
 - [Auto Numeric Node Expert Options](#)
 - [Auto Numeric node settings](#)
 - [Auto Cluster Node Model Options](#)
 - [Auto Cluster Node Expert Options](#)
 - [Auto Cluster Node Discard Options](#)
-

Evaluation Graphs

On the Model tab of the automated model nugget you can drill down to display individual graphs for each of the models shown. For Auto Classifier and Auto Numeric nuggets, the Graph tab displays both a graph and predictor importance that reflect the results of all the models combined. See the topic [Predictor Importance](#) for more information.

For Auto Classifier a distribution graph is shown, whereas a multiplot (also known as a scatterplot) is shown for Auto Numeric.

Related information

- [Automated Modeling Node Algorithm Settings](#)
 - [Automated Modeling Node Stopping Rules](#)
 - [Auto Classifier node](#)
 - [Auto Classifier node model options](#)
 - [Auto Classifier Node Expert Options](#)
 - [Auto Classifier Node Discard Options](#)
 - [Auto Classifier node settings](#)
 - [Automated Model Nuggets](#)
 - [Generating Nodes and Models](#)
 - [Generating Evaluation Charts](#)
 - [Auto Numeric node](#)
 - [Auto Numeric node model options](#)
 - [Auto Numeric Node Expert Options](#)
 - [Auto Numeric node settings](#)
-

Automated Model Nugget Summary

The Summary tab of the automated model nugget lists the fields used along with the build settings for each model, the total time elapsed to build all models, and the model ranking criterion used.

Continuous machine learning

As a result of IBM research, and inspired by natural selection in biology, *continuous machine learning* is available for the Auto Classifier node and the Auto Numeric node.

An inconvenience with modeling is models getting outdated due to changes to your data over time. This is commonly referred to as *model drift* or *concept drift*. To help overcome model drift effectively, SPSS Modeler provides continuous automated machine learning.

What is model drift? When you build a model based on historical data, it can become stagnant. In many cases, new data is always coming in—new variations, new patterns, new trends, etc.—that the old historical data doesn't capture. To solve this problem, IBM was inspired by the

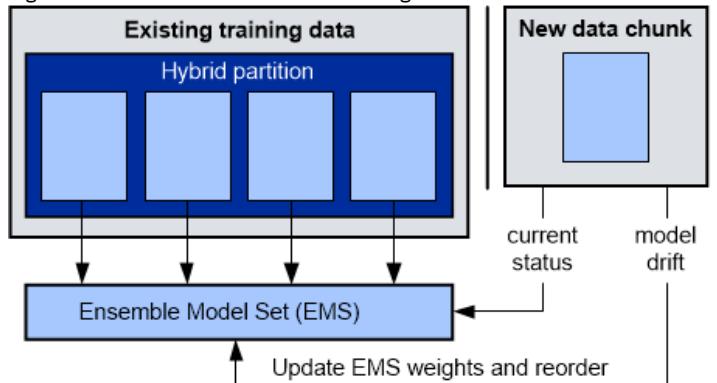
famous phenomenon in biology called the natural selection of species. Think of models as species and think of data as nature. Just as nature selects species, we should let data select the model. There's one big difference between models and species: species can evolve, but models are static once they've been built.

There are two preconditions for species to evolve; the first is gene mutation, and the second is population. Now, from a modeling perspective, to satisfy the first precondition (gene mutation), we should introduce new data changes into the existing model. To satisfy the second precondition (population), we should use a number of models rather than just one. What can represent a number of models? An Ensemble Model Set (EMS)!

The following figure illustrates how an EMS can evolve. The upper left portion of the figure represents historical data with hybrid partitions. The hybrid partitions ensure a rich initial EMS. The upper right portion of the figure represents a new chunk of data that becomes available, with vertical bars on each side. The left vertical bar represents current status, and the right vertical bar represents the status when there's a risk of model drift. In each new round of continuous machine learning, two steps are performed to evolve your model and avoid model drift.

First, you construct an ensemble model set (EMS) using existing training data. After that, when a new chunk of data becomes available, new models are built against that new data and added to the EMS as component models. The weights of existing component models in the EMS are reevaluated using the new data. As a result of this reevaluation, component models having higher weights are selected for the current prediction, and component models having lower weights may be deleted from the EMS. This process refreshes the EMS for both model weights and model instances, thus evolving in a flexible and efficient way to address the inevitable changes to your data over time.

Figure 1. Continuous auto machine learning



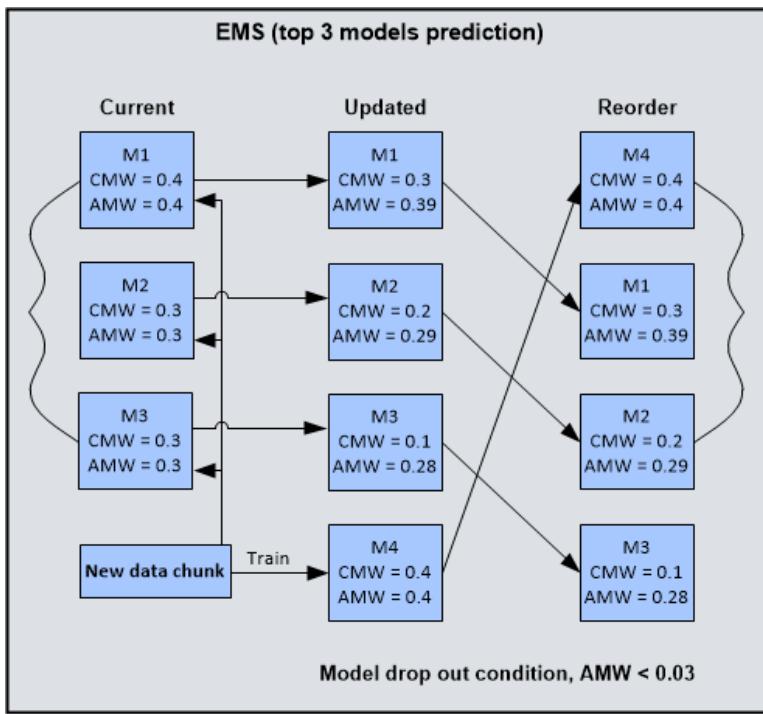
The ensemble model set (EMS) is a generated auto model nugget, and there's a refresh link between the auto modeling node and the generated auto model nugget that defines the refresh relationship between them. When you enable continuous auto machine learning, new data assets are continuously fed to auto modeling nodes to generate new component models. The model nugget is updated instead of replaced.

The following figure provides an example of the internal structure of an EMS in a continuous machine learning scenario. Only the top three component models are selected for the current prediction. For each component model (labeled as M1, M2, and M3), two kinds of weights are maintained. Current Model Weight (CMW) describes how a component model performs with a new chunk of data, and Accumulated Model Weight (AMW) describes the comprehensive performance of a component model against recent chunks of data. AMW is calculated iteratively via CMW and previous values of itself, and there's a hyper parameter beta to balance between them. The formula to calculate AMW is called *exponential moving average*.

When a new chunk of data becomes available, first SPSS Modeler uses it to build a few new component models. In this example figure, model four (M4) is built with CMW and AMW calculated during the initial model building process. Then SPSS Modeler uses the new chunk of data to reevaluate measures of existing component models (M1, M2, and M3) and update their CMW and AMW based on the reevaluation results. Finally, SPSS Modeler might reorder the component models based on CMW or AMW and select the top three component models accordingly.

In this figure, CMW is described using normalized value (sum = 1) and AMW is calculated based on CMW. In SPSS Modeler, the absolute value (equal to evaluation-weighted measure selected - for example, accuracy) is chosen to represent CMW and AMW for simplicity.

Figure 2. EMS structure



Note that there are two types of weights defined for each EMS component model as shown below, both of which could be used for selecting top N models and component model drop out:

- *Current Model Weight (CMW)* is computed via evaluation against the new data chunk (for example, evaluation accuracy on the new data chunk).
- *Accumulated Model Weight (AMW)* is computed via combining both CMW and existing AMW (for example, exponentially weighted moving average (EWMA)).

Exponential moving average formula for calculating AMW:

$$AMW = \beta * AMW + (1 - \beta) * CMW, \quad \text{suggested } \beta > 0.9$$

In SPSS Modeler, after running an Auto Classifier node to generate a model nugget, the following model options are available for continuous machine learning:

- Enable continuous auto machine learning during model refresh. Select this option to enable continuous machine learning. Keep in mind that consistent metadata (data model) must be used to train the continuous auto model. If you select this option, the other options below are enabled.
- Enable automatic model weights reevaluation. This option controls whether evaluation measures (accuracy, for example) are computed and updated during model refresh. If you select this option, an automatic evaluation process will run after the EMS (during model refresh). This is because it's usually necessary to reevaluate existing component models using new data to reflect the current state of your data. Then the weights of the EMS component models are assigned according to reevaluation results, and the weights are used to decide the proportion a component model contributes to the final ensemble prediction. This option is selected by default.

Figure 3. Model settings

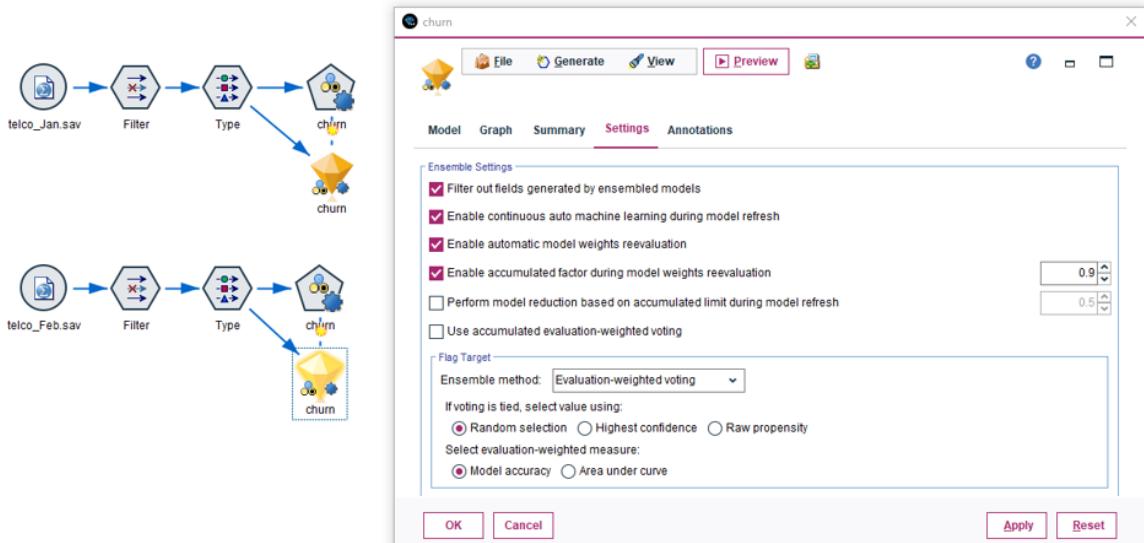


Figure 4. Flag target

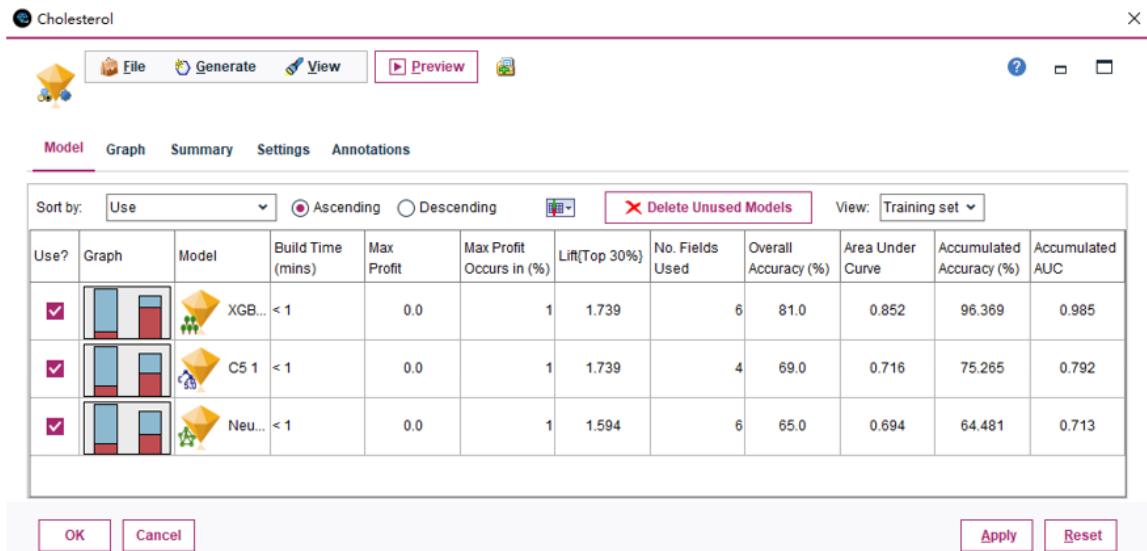
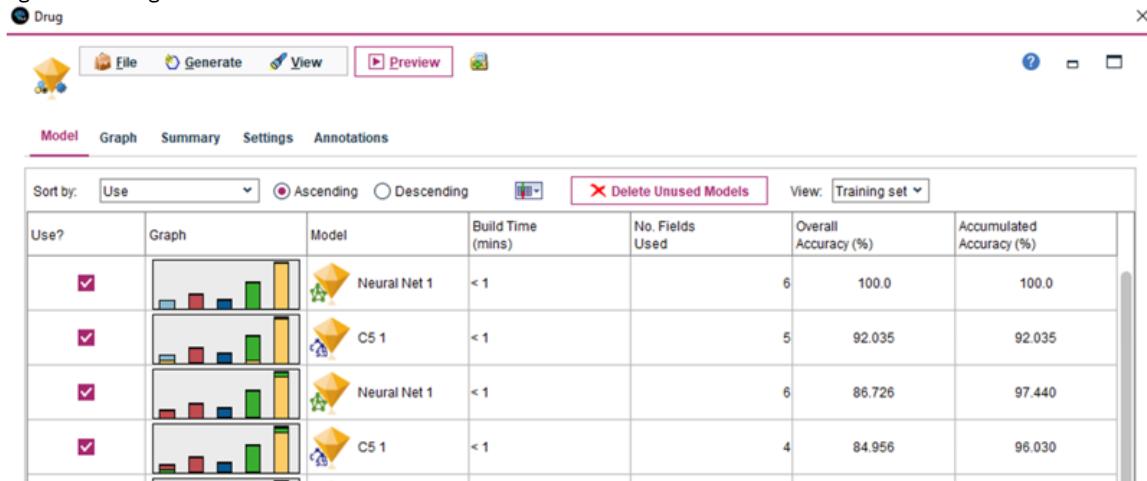


Figure 5. Set target



Following are the supported CMW and AMW for the Auto Classifier node:

Table 1. Supported CMW and AMW

Target type	CMW	AMW
flag target	Overall Accuracy Area Under Curve	Accumulated Accuracy Accumulated AUC
set target	Overall Accuracy	Accumulated Accuracy

The following three options are related to AMW, which is used to evaluate how a component model performs during recent data chunk periods:

- Enable accumulated factor during model weights reevaluation. If you select this option, AMW computation will be enabled during model weights reevaluation. AMW represents the comprehensive performance of an EMS component model during recent data chunk periods, related to the accumulated factor β defined in the AMW formula above, which you can adjust in the node properties. When this option isn't selected, only CMW will be computed. This option is selected by default.
- Perform model reduction based on accumulated limit during model refresh. Select this option if you want component models with an AMW value below the specified limit to be removed from the auto model EMS during model refresh. This can be helpful in discarding component models that are useless to prevent the auto model EMS from becoming too heavy. The accumulated limit value evaluation is related to the weighted measure used when Evaluation-weighted voting is selected as the ensemble method. See below.

Figure 6. Targets

Flag Target

Ensemble method: Evaluation-weighted voting

If voting is tied, select value using:

- Random selection
- Highest confidence
- Raw propensity

Select evaluation-weighted measure:

- Model Accuracy
- Area under curve

Set Target

Ensemble method: Confidence-weighted voting

If voting is tied, select value using:

- Random selection
- Highest confidence

Note that if you select Model Accuracy for the evaluation-weighted measure, models with an accumulated accuracy below the specified limit will be deleted. And if you select Area under curve for the evaluation-weighted measure, models with an accumulated AUC below the specified limit will be deleted.

By default, Model Accuracy is used for the evaluation-weighted measure for the Auto Classifier node, and there's an optional AUC ROC measure in the case of flag targets.

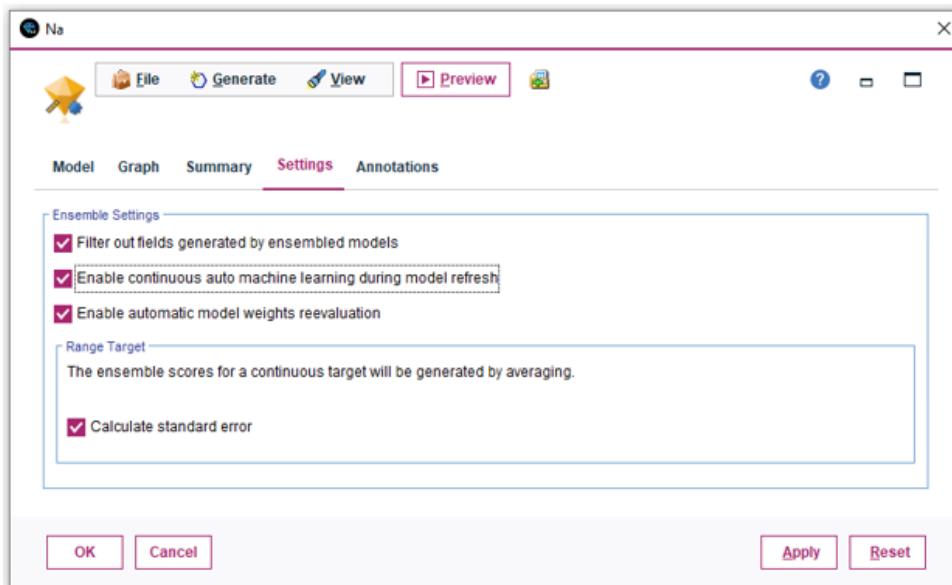
- Use accumulated evaluation-weighted voting. Select this option if you want AMW to be used for the current scoring/prediction. Otherwise, CMW will be used by default. This option is enabled when Evaluation-weighted voting is selected for the ensemble method. Note that for flag targets, by selecting this option, if you select Model Accuracy for the evaluation-weighted measure, then Accumulated Accuracy will be used as the AMW to perform the current scoring. Or if you select Area under curve for the evaluation-weighted measure, then Accumulated AUC will be used as the AMW to perform the current scoring. If you don't select this option and you select Model Accuracy for the evaluation-weighted measure, then Overall Accuracy will be used as the CMW to perform the current scoring. If you select Area under curve, Area under curve will be used as the CMW to perform the current scoring.

For set targets, if you select this Use accumulated evaluation-weighted voting option, then Accumulated Accuracy will be used as the AMW to perform the current scoring. Otherwise, Overall Accuracy will be used as the CMW to perform the current scoring.

With continuous auto machine learning, the auto model nugget is evolving all the time by rebuilding the auto model, which ensures that you get the most updated version reflecting the current state of your data. SPSS Modeler provides the flexibility for different top N component models in the EMS to be selected according to their current weights, which keeps pace with varying data during different periods.

Note: The Auto Numeric node is a much simpler case, providing a subset of the options in the Auto Classifier node.

Figure 7. Auto Numeric node

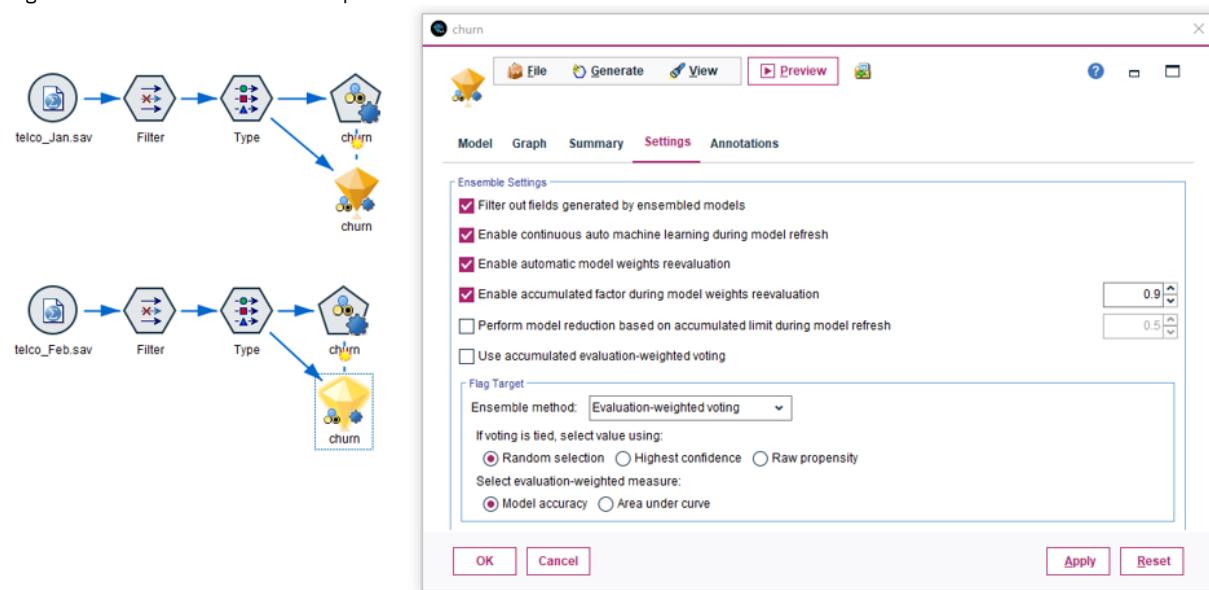


Example

In this example, continuous machine learning is used in the telecommunications industry to predict behavior and retain customers.

In the following flow, the data asset includes information about customers who left within the last month (**Churn** column). Since new data will be available every month, this scenario is suitable for continuous machine learning. In this example, January (**Jan**) data is used to construct an initial auto model, and then February (**Feb**) data is used to enhance the auto model via continuous machine learning.

Figure 8. Telecommunications example



In the upper branch of the flow, after the Data Asset node, there's a Filter node to filter out some unimportant fields. At the end of the branch, there's a terminal Auto Classifier modeling node. Under the node's expert settings, we select the algorithms we want to use for the training process. In this example, we select three algorithms: Logistic Regression, Bayesian Network, and Neural Network. Then we run the flow to generate an auto model nugget.

Now let's have a look at what's inside the auto model nugget. We can see it contains three component models for the three algorithms we selected. For each component model, there are several evaluation measures generated (such as accuracy and area under curve). These evaluation measures describe how a component model performs against training data (the January data set). You can select which component models to use in the current ensemble prediction.

Figure 9. Evaluations measures

churn

File Generate View Preview ? □

Model Graph Summary Settings Annotations

Sort by: Use Ascending Descending Delete Unused Models View: Training set

Us...	Graph	Model	Build Time (mins)	Max Profit	Max Profit Occurs in	Lift(Top 3...)	No. Fields Used	Overall Accuracy	Area Under Curve	Accumulated Accuracy (%)	Accumulated AUC
✓		Ba... < 1		205.0		13	2.222		80.6	0.852	80.6
✓		Lo... < 1		145.0		13	2.171		79.0	0.794	79.0
✓		N... < 1		145.0		11	2.041		79.2	0.791	79.2

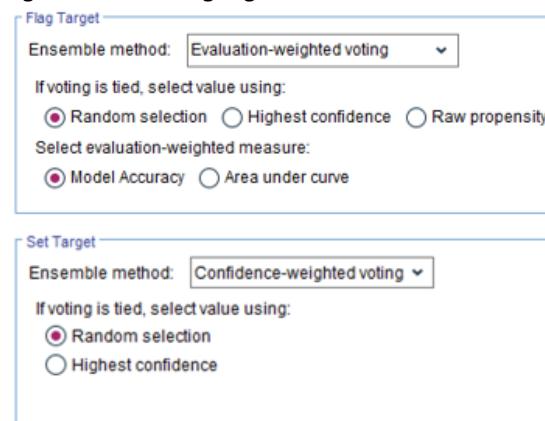
OK Cancel Apply Reset

In the upper branch of the flow, after the Data Asset node, there's a Filter node to filter out some unimportant fields. At the end of the branch, there's a terminal Auto Classifier modeling node. Under the node's expert settings, we select the algorithms we want to use for the training process. In this example, we select three algorithms: Logistic Regression, Bayesian Network, and Neural Network. Then we run the flow to generate an auto model nugget.

You may see accumulated evaluation measures, also. These accumulated measures are for continuous machine learning, as they describe how a component model performs with recent data changes, so that you're aware of the model's comprehensive performance over a period of time. As this is our initial auto model, we see that the initial values for the accumulated measures are the same as related current measures. By default, evaluation measures are calculated against training data, so there could be some degree of overfitting. To avoid this, the Auto Classifier node provides a build option that calculates more stable evaluation measures via cross validation.

Next, let's look at how the final ensemble prediction is generated. If we open an auto model's properties, under Ensemble Flag Targets, the training target churn field is a yes/no flag target. Under Ensemble Set Targets (for set target fields that contain more than two values), there's an Ensemble method drop-down. Several options are available in the drop-down (for example, Majority voting means each component model holds one ticket to vote, and Confidence-weighted voting means the confidence field of each component model's prediction is used as the voting weight—with higher confidence having more influence on the final ensemble prediction). Similarly, to enable better support for continuous machine learning, Evaluation-weighted voting is available so that the component model's evaluation measure (for example, model accuracy or area under curve) will be used as the voting weight. In the case of a flag target, there's also an option to select a specific evaluation measure as the voting weight when Evaluation-weighted voting is used. In the case of a set target, only Accuracy is currently supported.

Figure 10. Set and flag targets



Under the Ensemble Common settings is where you turn on continuous machine learning. Then we can use the February data to see what's happening. We might select two different algorithms to distinguish between the existing component model algorithms. Then after rebuilding the flow and viewing the auto model's content, we see that two new component models are added (C5 and C&RT). We also notice that the evaluation measures for the existing component models have been recalculated. Both CMW measures and AMW measures are different than before. We can now compare them to the corresponding measures in the original auto model.

Figure 11. Evaluation measures

Use?	Graph	Model	Build Time (mins)	Max Profit Occurs in (%)	Lift(Top 30%)	No. Fields Used	Overall Accuracy (%)	Area Under Curve	Accumulated Accuracy (%)	Accumulated AUC	
✓	[Bar Chart]	Bayesian N...	< 1	0.0	1	2.115	10	78.4	0.85	78.4	0.85
✓	[Bar Chart]	Logistic regres...	< 1	0.0	1	2.0	10	75.6	0.805	75.6	0.805
✓	[Bar Chart]	Neural Net 1	< 1	0.0	1	1.908	10	74.8	0.801	74.8	0.801
✓	[Bar Chart]	Neural Net 1	< 1	0.0	1	1.862	10	73.4	0.769	78.62	0.789
✓	[Bar Chart]	Logistic regres...	< 1	0.0	1	1.816	10	73.4	0.758	78.44	0.791
✓	[Bar Chart]	Bayesian N...	< 1	0.0	1	1.77	10	73.4	0.718	79.88	0.838

Now what? With the enhanced auto model, we can select a prioritized evaluation measure and get top-N component models ordered by that measure. Then we can use the top-N component models to participate in the final ensemble prediction for incoming predictive analytics requests. And if Evaluation-weighted voting is selected for the Ensemble method, we can use accumulated measures as voting weights by simply selecting the option Use accumulated evaluation-weighted voting under the Ensemble Common settings. If deselected, CMW measures will be used by default in evaluation-weighted voting.

With continuous machine learning, the auto model is evolving all the time as it is continuously rebuilding against new chunks of data, ensuring that your model is the most up-to-date version that reflects the current status of the data. This allows for the flexibility to select different top-N component models in the EMS according to their current or accumulated evaluation measures, to keep pace with varying data during different periods.

Decision Trees

- [Decision Tree Models](#)
- [The Interactive Tree Builder](#)
- [Building a Tree Model Directly](#)
- [Decision Tree Nodes](#)
- [C5.0 Node](#)
- [Tree-AS node](#)
- [Random Trees node](#)
- [C&R Tree, CHAID, QUEST, and C5.0 decision tree model nuggets](#)
- [C&R Tree, CHAID, QUEST, C5.0, and Apriori rule set model nuggets](#)

Decision Tree Models

Use decision tree models to develop classification systems that predict or classify future observations based on a set of decision rules. If you have data divided into classes that interest you (for example, high- versus low-risk loans, subscribers versus nonsubscribers, voters versus nonvoters, or types of bacteria), you can use your data to build rules that you can use to classify old or new cases with maximum accuracy. For example, you might build a tree that classifies credit risk or purchase intent based on age and other factors.

This approach, sometimes known as *rule induction*, has several advantages. First, the reasoning process behind the model is clearly evident when browsing the tree. This is in contrast to other *black box* modeling techniques in which the internal logic can be difficult to work out.

Second, the process automatically includes in its rule only the attributes that really matter in making a decision. Attributes that do not contribute to the accuracy of the tree are ignored. This can yield very useful information about the data and can be used to reduce the data to relevant fields before training another learning technique, such as a neural net.

Decision tree model nuggets can be converted into a collection of if-then rules (a *rule set*), which in many cases show the information in a more comprehensible form. The decision-tree presentation is useful when you want to see how attributes in the data can *split*, or *partition*, the population into subsets relevant to the problem. The Tree-AS node output is different from the other Decision tree nodes because it includes a list of rules directly in the nugget without having to create a rule set. The rule set presentation is useful if you want to see how particular groups of items relate to a specific conclusion. For example, the following rule provides a *profile* for a group of cars that is worth buying:

```
IF tested = 'yes'  
AND mileage = 'low'  
THEN -> 'BUY'.
```

Tree-building algorithms

Several algorithms are available for performing classification and segmentation analysis. These algorithms all perform basically the same thing, they examine all of the fields of your dataset to find the one that gives the best classification or prediction by splitting the data into subgroups. The process is applied recursively, splitting subgroups into smaller and smaller units until the tree is finished (as defined by certain stopping criteria). The target and input fields used in tree building can be continuous (numeric range) or categorical, depending on the algorithm used. If a continuous target is used, a regression tree is generated; if a categorical target is used, a classification tree is generated.

	The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered “pure” if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).
	The CHAID node generates decision trees using chi-square statistics to identify optimal splits. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.
	The QUEST node provides a binary classification method for building decision trees, designed to reduce the processing time required for large C&R Tree analyses while also reducing the tendency found in classification tree methods to favor inputs that allow more splits. Input fields can be numeric ranges (continuous), but the target field must be categorical. All splits are binary.
	The C5.0 node builds either a decision tree or a rule set. The model works by splitting the sample based on the field that

	provides the maximum information gain at each level. The target field must be categorical. Multiple splits into more than two subgroups are allowed.
	The Tree-AS node is similar to the existing CHAID node; however, the Tree-AS node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS® Modeler version 17. The node generates a decision tree by using chi-square statistics (CHAID) to identify optimal splits. This use of CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.
	The Random Trees node is similar to the existing C&RT node; however, the Random Trees node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS Modeler version 17. The Random Trees tree node generates a decision tree that you use to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered <i>pure</i> if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).

General uses of tree-based analysis

The following are some general uses of tree-based analysis:

Segmentation: Identify persons who are likely to be members of a particular class.

Stratification: Assign cases into one of several categories, such as high-, medium-, and low-risk groups.

Prediction: Create rules and use them to predict future events. Prediction can also mean attempts to relate predictive attributes to values of a continuous variable.

Data reduction and variable screening: Select a useful subset of predictors from a large set of variables for use in building a formal parametric model.

Interaction identification: Identify relationships that pertain only to specific subgroups and specify these in a formal parametric model.

Category merging and banding continuous variables: Recode group predictor categories and continuous variables with minimal loss of information.

Related information

- [C5.0 Node](#)

The Interactive Tree Builder

You can generate a tree model automatically, where the algorithm decides the best split at each level, or you can use the interactive tree builder to take control, applying your business knowledge to refine or simplify the tree before saving the model nugget.

1. Create a stream and add one of the decision tree nodes C&R Tree, CHAID, or QUEST.
Note: Interactive tree building is not supported for either Tree-AS or C5.0 trees.
2. Open the node and, on the Fields tab, select target and predictor fields and specify additional model options as needed. For specific instructions, see the documentation for each tree-building node.
3. On the Objectives panel of the Build Options tab, select Launch interactive session.
4. Click Run to launch the tree builder.

The current tree is displayed, starting with the root node. You can edit and prune the tree level-by-level and access gains, risks, and related information before generating one or more models.

Comments

- With the C&R Tree, CHAID, and QUEST nodes, any ordinal fields used in the model must have numeric storage (not string). If necessary, the Reclassify node can be used to convert them.
- Optionally, you can use a partition field to separate the data into training and test samples.
- As an alternative to using the tree builder, you can also generate a model directly from the modeling node as with other IBM® SPSS® Modeler models. See the topic [Building a Tree Model Directly](#) for more information.
- [Growing and Pruning the Tree](#)
- [Defining Custom Splits](#)
- [Split Details and Surrogates](#)

- [Customizing the Tree View](#)
- [Gains](#)
- [Risks](#)
- [Saving Tree Models and Results](#)
- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)

Related information

- [Growing and Pruning the Tree](#)
- [Defining Custom Splits](#)
- [Viewing Predictor Details](#)
- [Split Details and Surrogates](#)
- [Customizing the Tree View](#)
- [Gains](#)
- [Risks](#)
- [Saving Tree Models and Results](#)
- [Generating a Model from the Tree Builder](#)
- [Updating Tree Directives](#)
- [Exporting Model, Gain, and Risk Information](#)
- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)

Growing and Pruning the Tree

To launch the tree builder, execute a stream containing a C&R Tree, QUEST, or CHAID node, making sure the Launch interactive session option is selected on the Objectives panel of the Build Options tab.

The Viewer tab in the tree builder enables you to view the current tree, starting with the root node.

1. To grow the tree, from the menus choose:

Tree > Grow Tree

The system builds the tree by recursively splitting each branch until one or more stopping criteria are met. At each split, the best predictor is automatically selected based on the modeling method used.

2. Alternatively, select Grow Tree One Level to add a single level.
3. To add a branch below a specific node, select the node and select Grow Branch.
4. To choose the predictor used for a split, select the desired node and select Grow Branch with Custom Split. See the topic [Defining Custom Splits](#) for more information.
5. To prune a branch, select a node and select Remove Branch to clear up the selected node.
6. To remove the bottom level from the tree, select Remove One Level.
7. For C&R Tree and QUEST trees only, select Grow Tree and Prune to prune based on a cost-complexity algorithm that adjusts the risk estimate based on the number of terminal nodes, typically resulting in a simpler tree. See the topic [C&R Tree Node](#) for more information.

Reading Split Rules on the Viewer Tab

When viewing split rules on the Viewer tab, square brackets mean that the adjacent value is included in the range whereas parentheses indicate that the adjacent value is excluded from the range. The expression (23,37] therefore means from 23 exclusive to 37 inclusive; that is, from just above 23 to 37. On the Model tab, the same condition would be displayed as:

Age > 23 and Age <= 37

Interrupting tree growth. To interrupt a tree-growing operation (if it is taking longer than expected, for example), click the Stop Execution button on the toolbar.

Figure 1. Stop Execution button



The button is enabled only during tree growth. It stops the current growing operation at its current point, leaving any nodes that have already been added, without saving changes or closing the window. The tree builder remains open, enabling you to generate a model, update directives, or export output in the appropriate format, as needed.

Related information

- [The Interactive Tree Builder](#)
- [Defining Custom Splits](#)

- [Viewing Predictor Details](#)
 - [Split Details and Surrogates](#)
 - [Customizing the Tree View](#)
 - [Gains](#)
 - [Risks](#)
 - [Saving Tree Models and Results](#)
 - [Generating a Model from the Tree Builder](#)
 - [Updating Tree Directives](#)
 - [Exporting Model, Gain, and Risk Information](#)
 - [Generating Filter and Select Nodes](#)
 - [Generating a Rule Set from a Decision Tree](#)
-

Defining Custom Splits

The Define Split dialog box enables you to select the predictor and specify conditions for each split.

1. In the tree builder, select a node on the Viewer tab, and from the menus choose:
Tree > Grow Branch with Custom Split
2. Select the desired predictor from the drop-down list, or click on the Predictors button to view details of each predictor. See the topic [Viewing Predictor Details](#) for more information.
3. You can accept the default conditions for each split or select Custom to specify conditions for the split as appropriate.
 - For continuous (numeric range) predictors, you can use the Edit Range Values fields to specify the range of values that fall into each new node.
 - For categorical predictors, you can use the Edit Set Values or Edit Ordinal Values fields to specify the specific values (or range of values in case of an ordinal predictor) that map to each new node.
4. Select Grow to regrow the branch using the selected predictor.

The tree can generally be split using any predictor, regardless of stopping rules. The only exceptions are when the node is pure (meaning that 100% of cases fall into the same target class, thus there is nothing left to split) or the chosen predictor is constant (there is nothing to split against).

Missing values into. For CHAID trees only, if missing values are available for a given predictor, you have the option when defining a custom split to assign them to a specific child node. (With C&R Tree and QUEST, missing values are handled using surrogates as defined in the algorithm. See the topic [Split Details and Surrogates](#) for more information.)

- [Viewing Predictor Details](#)

Related information

- [The Interactive Tree Builder](#)
 - [Growing and Pruning the Tree](#)
 - [Viewing Predictor Details](#)
 - [Split Details and Surrogates](#)
 - [Customizing the Tree View](#)
 - [Gains](#)
 - [Risks](#)
 - [Saving Tree Models and Results](#)
 - [Generating a Model from the Tree Builder](#)
 - [Updating Tree Directives](#)
 - [Exporting Model, Gain, and Risk Information](#)
 - [Generating Filter and Select Nodes](#)
 - [Generating a Rule Set from a Decision Tree](#)
-

Viewing Predictor Details

The Select Predictor dialog box displays statistics on available predictors (or "competitors" as they are sometimes called) that can be used for the current split.

- For CHAID and exhaustive CHAID, the chi-square statistic is listed for each categorical predictor; if a predictor is a numeric range, the F statistic is shown. The chi-square statistic is a measure of how independent the target field is from the splitting field. A high chi-square statistic generally relates to a lower probability, meaning that there is less chance that the two fields are independent—an indication that the split is a good one. Degrees of freedom are also included because these take into account the fact that it is easier for a three-way split to have a large statistic and small probability than it is for a two-way split.

- For C&R Tree and QUEST, the improvement for each predictor is displayed. The greater the improvement, the greater the reduction in impurity between the parent and child nodes if that predictor is used. (A pure node is one in which all cases fall into a single target category; the lower the impurity across the tree, the better the model fits the data.) In other words, a high improvement figure generally indicates a useful split for this type of tree. The impurity measure used is specified in the tree-building node.

Related information

- [The Interactive Tree Builder](#)
- [Growing and Pruning the Tree](#)
- [Defining Custom Splits](#)
- [Split Details and Surrogates](#)
- [Customizing the Tree View](#)
- [Gains](#)
- [Risks](#)
- [Saving Tree Models and Results](#)
- [Generating a Model from the Tree Builder](#)
- [Updating Tree Directives](#)
- [Exporting Model, Gain, and Risk Information](#)
- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)

Split Details and Surrogates

You can select any node on the Viewer tab and select the split information button on the right side of the toolbar to view details about the split for that node. The split rule used is displayed, along with relevant statistics. For C&R Tree categorical trees, improvement and association are displayed. The association is a measure of correspondence between a surrogate and the primary split field, with the “best” surrogate generally being the one that most closely mimics the split field. For C&R Tree and QUEST, any surrogates used in place of the primary predictor are also listed.

To edit the split for the selected node, you can click the icon on the left side of the surrogates panel to open the Define Split dialog box. (As a shortcut, you can select a surrogate from the list before clicking the icon to select it as the primary split field.)

Surrogates. Where applicable, any surrogates for the primary split field are shown for the selected node. Surrogates are alternate fields used if the primary predictor value is missing for a given record. The maximum number of surrogates allowed for a given split is specified in the tree-building node, but the actual number depends on the training data. In general, the more missing data, the more surrogates are likely to be used. For other decision tree models, this tab is empty.

Note: To be included in the model, surrogates must be identified during the training phase. If the training sample has no missing values, then no surrogates will be identified, and any records with missing values encountered during testing or scoring will automatically fall into the child node with the largest number of records. If missing values are expected during testing or scoring, be sure that values are missing from the training sample, as well. Surrogates are not available for CHAID trees.

Although surrogates are not used for CHAID trees, when defining a custom split you have the option to assign them to a specific child node. See the topic [Defining Custom Splits](#) for more information.

Related information

- [The Interactive Tree Builder](#)
- [Growing and Pruning the Tree](#)
- [Defining Custom Splits](#)
- [Viewing Predictor Details](#)
- [Customizing the Tree View](#)
- [Gains](#)
- [Risks](#)
- [Saving Tree Models and Results](#)
- [Generating a Model from the Tree Builder](#)
- [Updating Tree Directives](#)
- [Exporting Model, Gain, and Risk Information](#)
- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)

Customizing the Tree View

The Viewer tab in the tree builder displays the current tree. By default, all branches in the tree are expanded, but you can expand and collapse branches and customize other settings as needed.

- Click the minus sign (-) at the bottom right corner of a parent node to hide all of its child nodes. Click the plus sign (+) at the bottom right corner of a parent node to display its child nodes.
- Use the View menu or the toolbar to change the orientation of the tree (top-down, left-to-right, or right-to-left).
- Click the "Display field and value labels" button on the main toolbar to show or hide field and value labels.
- Use the magnifying glass buttons to zoom the view in or out, or click the tree map button on the right side of the toolbar to view a diagram of the complete tree.
- If a partition field is in use, you can swap the tree view between training and testing partitions (View > Partition). When the testing sample is displayed, the tree can be viewed but not edited. (The current partition is displayed in the status bar in the lower right corner of the window.)
- Click the split information button (the "i" button on the far right of the toolbar) to view details on the current split. See the topic [Split Details and Surrogates](#) for more information.
- Display statistics, graphs, or both within each node (see below).

Displaying Statistics and Graphs

Node statistics. For a categorical target field, the table in each node shows the number and percentage of records in each category and the percentage of the entire sample that the node represents. For a continuous (numeric range) target field, the table shows the mean, standard deviation, number of records, and predicted value of the target field.

Node graphs. For a categorical target field, the graph is a bar chart of percentages in each category of the target field. Preceding each row in the table is a color swatch that corresponds to the color that represents each of the target field categories in the graphs for the node. For a continuous (numeric range) target field, the graph shows a histogram of the target field for records in the node.

Related information

- [The Interactive Tree Builder](#)
 - [Growing and Pruning the Tree](#)
 - [Defining Custom Splits](#)
 - [Viewing Predictor Details](#)
 - [Split Details and Surrogates](#)
 - [Gains](#)
 - [Risks](#)
 - [Saving Tree Models and Results](#)
 - [Generating a Model from the Tree Builder](#)
 - [Updating Tree Directives](#)
 - [Exporting Model, Gain, and Risk Information](#)
 - [Generating Filter and Select Nodes](#)
 - [Generating a Rule Set from a Decision Tree](#)
-

Gains

The Gains tab displays statistics for all terminal nodes in the tree. Gains provide a measure of how far the mean or proportion at a given node differs from the overall mean. Generally speaking, the greater this difference, the more useful the tree is as a tool for making decisions. For example, an index or "lift" value of 148% for a node indicates that records in the node are about one-and-a-half times as likely to fall under the target category as for the dataset as a whole.

For C&R Tree and QUEST nodes where an overfit prevention set is specified, two sets of statistics are displayed:

- tree growing set - the training sample with the overfit prevention set removed
- overfit prevention set

For other C&R Tree and QUEST interactive trees, and for all CHAID interactive trees, only the tree growing set statistics are displayed.

The Gains tab enables you to:

- Display node-by-node, cumulative, or quantile statistics.
- Display gains or profits.
- Swap the view between tables and charts.
- Select the target category (categorical targets only).
- Sort the table in ascending or descending order based on the index percentage. If statistics for multiple partitions are displayed, sorts are always applied on the training sample rather than on the testing sample.

In general, selections made in the gains table will be updated in the tree view and vice versa. For example, if you select a row in the table, the corresponding node will be selected in the tree.

- [Classification Gains](#)
- [Classification Profits and ROI](#)
- [Regression Gains](#)
- [Gains Charts](#)
- [Gains-Based Selection](#)

Related information

- [The Interactive Tree Builder](#)
- [Growing and Pruning the Tree](#)
- [Defining Custom Splits](#)
- [Viewing Predictor Details](#)
- [Split Details and Surrogates](#)
- [Customizing the Tree View](#)
- [Risks](#)
- [Saving Tree Models and Results](#)
- [Generating a Model from the Tree Builder](#)
- [Updating Tree Directives](#)
- [Exporting Model, Gain, and Risk Information](#)
- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)
- [Classification Gains](#)
- [Classification Profits and ROI](#)
- [Regression Gains](#)
- [Gains Charts](#)
- [Gains-Based Selection](#)

Classification Gains

For classification trees (those with a categorical target variable), the gain index percentage tells you how much greater the proportion of a given target category at each node differs from the overall proportion.

Node-by-Node Statistics

In this view, the table displays one row for each terminal node. For example, if the overall response to your direct mail campaign was 10% but 20% of the records that fall into node X responded positively, the index percentage for the node would be 200%, indicating that respondents in this group are twice as likely to buy relative to the overall population.

For C&R Tree and QUEST nodes where an overfit prevention set is specified, two sets of statistics are displayed:

- tree growing set - the training sample with the overfit prevention set removed
- overfit prevention set

For other C&R Tree and QUEST interactive trees, and for all CHAID interactive trees, only the tree growing set statistics are displayed.

Nodes. The ID of the current node (as displayed on the Viewer tab).

Node: n. The total number of records in that node.

Node (%). The percentage of all records in the dataset that fall into this node.

Gain: n. The number of records with the selected target category that fall into this node. In other words, of all the records in the dataset that fall under the target category, how many are in this node?

Gain (%). The percentage of all records in the target category, across the entire dataset, that fall into this node.

Response (%). The percentage of records in the current node that fall under the target category. Responses in this context are sometimes referred to as "hits."

Index (%). The response percentage for the current node expressed as a percentage of the response percentage for the entire dataset. For example, an index value of 300% indicates that records in this node are three times as likely to fall under the target category as for the dataset as a whole.

Cumulative Statistics

In the cumulative view, the table displays one node per row, but statistics are cumulative, sorted in ascending or descending order by index percentage. For example if a descending sort is applied, the node with the highest index percentage is listed first, and statistics in the rows that follow are cumulative for that row and above.

The cumulative index percentage decreases row-by-row as nodes with lower and lower response percentages are added. The cumulative index for the final row is always 100% because at this point the entire dataset is included.

Quantiles

In this view, each row in the table represents a quantile rather than a node. The quantiles are either quartiles, quintiles (fifths), deciles (tenths), vigintiles (twentieths), or percentiles (hundredths). Multiple nodes can be listed in a single quantile if more than one node is needed to make up that percentage (for example, if quartiles are displayed but the top two nodes contain fewer than 50% of all cases). The rest of the table is cumulative and can be interpreted in the same manner as the cumulative view.

Related information

- [Gains](#)
 - [Classification Profits and ROI](#)
 - [Regression Gains](#)
 - [Gains Charts](#)
 - [Gains-Based Selection](#)
 - [Risks](#)
-

Classification Profits and ROI

For classification trees, gains statistics can also be displayed in terms of profit and ROI (return on investment). The Define Profits dialog box enables you to specify revenue and expenses for each category.

1. On the Gains tab, click the Profit button (labeled \$/\$) on the toolbar to access the dialog box.
2. Enter revenue and expense values for each category of the target field.

For example, if it costs you \$0.48 to mail an offer to each customer and the revenue from a positive response is \$9.95 for a three-month subscription, then each *no* response costs you \$0.48 and each *yes* earns you \$9.47 (calculated as 9.95–0.48).

In the gains table, **profit** is calculated as the sum of revenues minus expenditures for each of the records at a terminal node. **ROI** is total profit divided by total expenditure at a node.

Comments

- Profit values affect only average profit and ROI values displayed in the gains table, as a way of viewing statistics in terms more applicable to your bottom line. They do not affect the basic tree model structure. Profits should not be confused with misclassification costs, which are specified in the tree-building node and are factored into the model as a way of protecting against costly mistakes.
- Profit specifications are not persisted between one interactive tree-building session and the next.

Related information

- [Gains](#)
 - [Classification Gains](#)
 - [Regression Gains](#)
 - [Gains Charts](#)
 - [Gains-Based Selection](#)
 - [Risks](#)
-

Regression Gains

For regression trees, you can choose between node-by-node, cumulative node-by-node, and quantile views. Average values are shown in the table. Charts are available only for quantiles.

Related information

- [Gains](#)
- [Classification Gains](#)
- [Classification Profits and ROI](#)
- [Gains Charts](#)
- [Gains-Based Selection](#)
- [Risks](#)

Gains Charts

Charts can be displayed on the Gains tab as an alternative to tables.

1. On the Gains tab, select the Quantiles icon (third from left on the toolbar). (Charts are not available for node-by-node or cumulative statistics.)
2. Select the Charts icon.
3. Select the displayed units (percentiles, deciles, and so on) from the drop-down list as desired.
4. Select Gains, Response, or Lift to change the displayed measure.

Gains Chart

The gains chart plots the values in the *Gains (%)* column from the table. Gains are defined as the proportion of hits in each increment relative to the total number of hits in the tree, using the equation:

$$(\text{hits in increment} / \text{total number of hits}) \times 100\%$$

The chart effectively illustrates how widely you need to cast the net to capture a given percentage of all the hits in the tree. The diagonal line plots the expected response for the entire sample, if the model were not used. In this case, the response rate would be constant, since one person is just as likely to respond as another. To double your yield, you would need to ask twice as many people. The curved line indicates how much you can improve your response by including only those who rank in the higher percentiles based on gain. For example, including the top 50% might net you more than 70% of the positive responses. The steeper the curve, the higher the gain.

Lift Chart

The lift chart plots the values in the *Index (%)* column in the table. This chart compares the percentage of records in each increment that are hits with the overall percentage of hits in the training dataset, using the equation:

$$(\text{hits in increment} / \text{records in increment}) / (\text{total number of hits} / \text{total number of records})$$

Response Chart

The response chart plots the values in the *Response (%)* column of the table. The response is a percentage of records in the increment that are hits, using the equation:

$$(\text{responses in increment} / \text{records in increment}) \times 100\%$$

Related information

- [Gains](#)
- [Classification Gains](#)
- [Classification Profits and ROI](#)
- [Regression Gains](#)
- [Gains-Based Selection](#)
- [Risks](#)

Gains-Based Selection

The Gains-Based Selection dialog box enables you to automatically select terminal nodes with the best (or worst) gains based on a specified rule or threshold. You can then generate a Select node based on the selection.

1. On the Gains tab, select the node-by-node or cumulative view and select the target category on which you want to base the selection. (Selections are based on the current table display and are not available for quantiles.)
2. On the Gains tab, from the menus choose:
Edit...>Select Terminal Nodes...>Gains-Based Selection

Select only. You can select matching nodes or nonmatching nodes—for example, to select *all but* the top 100 records.

Match by gains information. Matches nodes based on gain statistics for the current target category, including:

- Nodes where the gain, response, or lift (index) matches a specified threshold—for example, response greater than or equal to 50%.
 - The top *n* nodes based on the gain for the target category.
 - The top nodes up to a specified number of records.
 - The top nodes up to a specified percentage of training data.
3. Click OK to update the selection on the Viewer tab.
 4. To create a new Select node based on the current selection on the Viewer tab, choose Select Node from the Generate menu. See the topic [Generating Filter and Select Nodes](#) for more information.

Note: Since you are actually selecting nodes rather than records or percentages, a perfect match with the selection criterion may not always be achieved. The system selects complete nodes *up to* the specified level. For example, if you select the top 12 cases and you have 10 in the first node and two in the second node, only the first node will be selected.

Related information

- [Gains](#)
 - [Classification Gains](#)
 - [Classification Profits and ROI](#)
 - [Regression Gains](#)
 - [Gains Charts](#)
 - [Risks](#)
-

Risks

Risks tell you the chances of misclassification at any level. The Risks tab displays a point risk estimate and (for categorical outputs) a misclassification table.

- For numeric predictions, the risk is a pooled estimate of the variance at each of the terminal nodes.
- For categorical predictions, the risk is the proportion of cases incorrectly classified, adjusted for any priors or misclassification costs.

Related information

- [The Interactive Tree Builder](#)
 - [Growing and Pruning the Tree](#)
 - [Defining Custom Splits](#)
 - [Viewing Predictor Details](#)
 - [Split Details and Surrogates](#)
 - [Customizing the Tree View](#)
 - [Gains](#)
 - [Saving Tree Models and Results](#)
 - [Generating a Model from the Tree Builder](#)
 - [Updating Tree Directives](#)
 - [Exporting Model, Gain, and Risk Information](#)
 - [Generating Filter and Select Nodes](#)
 - [Generating a Rule Set from a Decision Tree](#)
 - [Classification Gains](#)
 - [Classification Profits and ROI](#)
 - [Regression Gains](#)
 - [Gains Charts](#)
 - [Gains-Based Selection](#)
-

Saving Tree Models and Results

You can save or export the results of your interactive tree-building sessions in a number of ways, including:

- Generate a model based on the current tree (Generate > Generate model).
- Save the directives used to grow the current tree. The next time the tree-building node is executed, the current tree will automatically be regrown, including any custom splits that you have defined.
- Export model, gain, and risk information. See the topic [Exporting Model, Gain, and Risk Information](#) for more information.

From either the tree builder or a tree model nugget, you can:

- Generate a Filter or Select node based on the current tree. See [Generating Filter and Select Nodes](#) for more information.
- Generate a Rule Set nugget that represents the tree structure as a set of rules defining the terminal branches of the tree. See [Generating a Rule Set from a Decision Tree](#) for more information.
- In addition, for tree model nuggets only, you can export the model in PMML format. See [The models palette](#) for more information. If the model includes any custom splits, this information is not preserved in the exported PMML. (The split is preserved, but the fact that it is custom rather than chosen by the algorithm is not.)
- Generate a graph based on a selected part of the current tree. Note that this only works for a nugget when it is attached to other nodes in a stream. See [Generating Graphs](#) for more information.

Note: The interactive tree itself cannot be saved. To avoid losing your work, generate a model and/or update tree directives before closing the tree builder window.

- [Generating a Model from the Tree Builder](#)
- [Tree-Growing Directives](#)
- [Updating Tree Directives](#)
- [Exporting Model, Gain, and Risk Information](#)

Related information

- [The Interactive Tree Builder](#)
- [Growing and Pruning the Tree](#)
- [Defining Custom Splits](#)
- [Viewing Predictor Details](#)
- [Split Details and Surrogates](#)
- [Customizing the Tree View](#)
- [Gains](#)
- [Risks](#)
- [Generating a Model from the Tree Builder](#)
- [Updating Tree Directives](#)
- [Exporting Model, Gain, and Risk Information](#)
- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)

Generating a Model from the Tree Builder

To generate a model based on the current tree, from the tree builder menus choose:

Generate...> Model

In the Generate New Model dialog box you can choose from the following options:

Model name. You can specify a custom name or generate the name automatically based on the name of the modeling node.

Create node on. You can add the node on the Canvas, GM Palette, or Both.

Include tree directives. To include the directives from the current tree in the generated model, select this box. This enables you to regenerate the tree, if required. See the topic [Tree-Growing Directives](#) for more information.

Related information

- [The Interactive Tree Builder](#)
- [Growing and Pruning the Tree](#)
- [Defining Custom Splits](#)
- [Viewing Predictor Details](#)
- [Split Details and Surrogates](#)
- [Customizing the Tree View](#)
- [Gains](#)
- [Risks](#)
- [Saving Tree Models and Results](#)
- [Updating Tree Directives](#)
- [Exporting Model, Gain, and Risk Information](#)
- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)

Tree-Growing Directives

For C&R Tree, CHAID, and QUEST models, tree directives specify conditions for growing the tree, one level at a time. Directives are applied each time the interactive tree builder is launched from the node.

- Directives are most safely used as a way to regenerate a tree created during a previous interactive session. See the topic [Updating Tree Directives](#) for more information. You can also edit directives manually, but this should be done with care.

- Directives are highly specific to the structure of the tree they describe. Thus, any change to the underlying data or modeling options may cause a previously valid set of directives to fail. For example, if the CHAID algorithm changes a two-way split to a three-way split based on updated data, any directives based on the previous two-way split would fail.

Note: If you choose to generate a model directly (without using the tree builder), any tree directives are ignored.

Editing Directives

1. To view or edit saved directives, open the tree-building node and select the Objective panel of the Build Options tab.
2. Select Launch interactive session to enable the controls, select Use tree directives, and click Directives.

Directive Syntax

Directives specify conditions for growing the tree, starting with the root node. For example to grow the tree one level:

```
Grow Node Index 0 Children 1 2
```

Since no predictor is specified, the algorithm chooses the best split.

Note that the first split must always be on the root node (**Index 0**) and the index values for both children must be specified (1 and 2 in this case). It is invalid to specify **Grow Node Index 2 Children 3 4** unless you first grew the root that created Node 2.

To grow the tree:

```
Grow Tree
```

To grow and prune the tree (C&R Tree only):

```
Grow And Prune Tree
```

To specify a custom split for a continuous predictor:

```
Grow Node Index 0 Children 1 2 Spliton
  ("EDUCATE", Interval ( NegativeInfinity, 12.5)
   Interval ( 12.5, Infinity ))
```

To split on a nominal predictor with two values:

```
Grow Node Index 2 Children 3 4 Spliton
  ("GENDER", Group( "0.0" )Group( "1.0" ))
```

To split on a nominal predictor with multiple values:

```
Grow Node Index 6 Children 7 8 Spliton
  ("ORGs", Group( "2.0","4.0" )
   Group( "0.0","1.0","3.0","6.0" ))
```

To split on an ordinal predictor:

```
Grow Node Index 4 Children 5 6 Spliton
  ("CHILDS", Interval ( NegativeInfinity, 1.0)
   Interval ( 1.0, Infinity ))
```

Note: When specifying custom splits, field names and values (**EDUCATE**, **GENDER**, **CHILDS**, etc.) are case sensitive.

Directives for CHAID Trees

Directives for CHAID trees are particularly sensitive to changes in the data or model because--unlike C&R Tree and QUEST--they are not constrained to use binary splits. For example, the following syntax looks perfectly valid but would fail if the algorithm splits the root node into more than two children:

```
Grow Node Index 0 Children 1 2
Grow Node Index 1 Children 3 4
```

With CHAID, it is possible that Node 0 will have 3 or 4 children, which would cause the second line of syntax to fail.

Using Directives in Scripts

Directives can also be embedded in scripts using triple quotation marks.

Related information

- [C&R Tree Node](#)
- [QUEST Node](#)

Updating Tree Directives

To preserve your work from an interactive tree-building session, you can save the directives used to generate the current tree. Unlike saving a model nugget, which cannot be edited further, this enables you to regenerate the tree in its current state for further editing.

To update directives, from the tree builder menus choose:

File>Update Directives

Directives are saved in the modeling node used to create the tree (either C&R Tree, QUEST, or CHAID) and can be used to regenerate the current tree. See the topic [Tree-Growing Directives](#) for more information.

Related information

- [The Interactive Tree Builder](#)
- [Growing and Pruning the Tree](#)
- [Defining Custom Splits](#)
- [Viewing Predictor Details](#)
- [Split Details and Surrogates](#)
- [Customizing the Tree View](#)
- [Gains](#)
- [Risks](#)
- [Saving Tree Models and Results](#)
- [Generating a Model from the Tree Builder](#)
- [Exporting Model, Gain, and Risk Information](#)
- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)

Exporting Model, Gain, and Risk Information

From the tree builder, you can export model, gain, and risk statistics in text, HTML, or image formats as appropriate.

1. In the tree builder window, select the tab or view that you want to export.
2. From the menus choose:
File>Export
3. Select Text, HTML, or Graph as appropriate, and select the specific items you want to export from the submenu.

Where applicable, the export is based on current selections.

Exporting Text or HTML formats. You can export gain or risk statistics for the training or testing partition (if defined). The export is based on the current selections on the Gains tab—for example, you can choose node-by-node, cumulative, or quantile statistics.

Exporting graphics. You can export the current tree as displayed on the Viewer tab or export gains charts for the training or testing partition (if defined). Available formats include .JPEG, .PNG, and .BMP. For gains, the export is based on current selections on the Gains tab (available only when a chart is displayed).

Related information

- [The Interactive Tree Builder](#)
- [Growing and Pruning the Tree](#)
- [Defining Custom Splits](#)
- [Viewing Predictor Details](#)
- [Split Details and Surrogates](#)
- [Customizing the Tree View](#)
- [Gains](#)
- [Risks](#)
- [Saving Tree Models and Results](#)
- [Generating a Model from the Tree Builder](#)
- [Upgrading Tree Directives](#)
- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)

Generating Filter and Select Nodes

In the tree builder window, or when browsing a decision tree model nugget, from the menus choose:

Generate > Filter Node

or

> Select Node

Filter Node. Generates a node that filters any fields not used by the current tree. This is a quick way to pare down the dataset to include only those fields that are selected as important by the algorithm. If there is a Type node upstream from this decision tree node, any fields with the role *Target* are passed on by the Filter model nugget.

Select Node. Generates a node that selects all records that fall into the current node. This option requires that one or more tree branches be selected on the Viewer tab.

The model nugget is placed on the stream canvas.

Related information

- [The Interactive Tree Builder](#)
- [Growing and Pruning the Tree](#)
- [Defining Custom Splits](#)
- [Viewing Predictor Details](#)
- [Split Details and Surrogates](#)
- [Customizing the Tree View](#)
- [Gains](#)
- [Risks](#)
- [Saving Tree Models and Results](#)
- [Generating a Model from the Tree Builder](#)
- [Updating Tree Directives](#)
- [Exporting Model, Gain, and Risk Information](#)
- [Generating a Rule Set from a Decision Tree](#)
- [C&R Tree, CHAID, QUEST, and C5.0 decision tree model nuggets](#)
- [Decision Tree Model Rules](#)
- [Decision Tree Model Viewer](#)
- [Decision Tree/Rule Set model nugget settings](#)
- [Boosted C5.0 Models](#)
- [Generating Graphs](#)

Generating a Rule Set from a Decision Tree

You can generate a Rule Set model nugget that represents the tree structure as a set of rules that define the terminal branches of the tree. Rule sets can often retain most of the important information from a full decision tree but with a less complex model. The most important difference is that with a rule set, more than one rule might apply for any particular record or no rules at all might apply. For example, you might see all of the rules that predict a *no* outcome followed by all of the rules that predict *yes*. If multiple rules apply, each rule gets a weighted "vote" based on the confidence that is associated with that rule, and the final prediction is decided by combining the weighted votes of all of the rules that apply to the record in question. If no rule applies, a default prediction is assigned to the record.

Note: When you score a rule set you might notice differences in scoring compared to scoring against the tree; this is because each terminal branch in a tree is scored independently. One area where this difference might become noticeable is when there are missing values in your data. Rule sets can be generated only from trees with categorical target fields (no regression trees).

In the tree builder window, or when you browse a decision tree model nugget, from the menus choose:

Generate > Rule Set

Rule set name Specify the name of the new Rule Set model nugget.

Create node on Controls the location of the new Rule Set model nugget. Select Canvas, GM Palette, or Both.

Minimum instances Specify the minimum number of instances (number of records to which the rule applies) to preserve in the Rule Set model nugget. Rules with support less than the specified value are not included in the new rule set.

Minimum confidence Specify the minimum confidence for rules to be preserved in the Rule Set model nugget. Rules with confidence less than the specified value are not included in the new rule set.

Related information

- [The Interactive Tree Builder](#)
- [Growing and Pruning the Tree](#)
- [Defining Custom Splits](#)
- [Viewing Predictor Details](#)
- [Split Details and Surrogates](#)
- [Customizing the Tree View](#)
- [Gains](#)
- [Risks](#)
- [Saving Tree Models and Results](#)
- [Generating a Model from the Tree Builder](#)
- [Updating Tree Directives](#)
- [Exporting Model, Gain, and Risk Information](#)
- [Generating Filter and Select Nodes](#)
- [C&R Tree, CHAID, QUEST, and C5.0 decision tree model nuggets](#)
- [Decision Tree Model Rules](#)
- [Decision Tree Model Viewer](#)
- [Decision Tree/Rule Set model nugget settings](#)
- [Boosted C5.0 Models](#)
- [Generating Graphs](#)

Building a Tree Model Directly

As an alternative to using the interactive tree builder, you can build a decision tree model directly from the node when the stream is run. This is consistent with most other model-building nodes. For C5.0 tree and Tree-AS models, which are not supported by the interactive tree builder, this is the only method that can be used.

1. Create a stream and add one of the decision tree nodes—C&R Tree, CHAID, QUEST, C5.0, or Tree-AS.
2. For C&R Tree, QUEST or CHAID, on the Objective panel of the Build Options tab, choose one of the main objectives. If you choose Build a single tree, ensure that Mode is set to Generate model.
For C5.0, on the Model tab, set Output type to Decision tree.
For Tree-AS, on the Basics panel of the Build Options tab, select the Tree growing algorithm type.
3. Select target and predictor fields and specify additional model options, as needed. For specific instructions, see the documentation for each tree-building node.
4. Run the stream to generate the model.

Comments about tree building

- When generating trees using this method, tree-growing directives are ignored.
- Whether interactive or direct, both methods of creating decision trees ultimately generate similar models. It's just a question of how much control you want along the way.

Decision Tree Nodes

The Decision Tree nodes in IBM® SPSS® Modeler provide access to the following tree-building algorithms:

- C&R Tree
- QUEST
- CHAID
- C5.0
- Tree-AS
- Random Trees

See the topic [Decision Tree Models](#) for more information.

The algorithms are similar in that they can all construct a decision tree by recursively splitting the data into smaller and smaller subgroups. However, there are some important differences.

Input fields. The input fields (predictors) can be any of the following types (measurement levels): continuous, categorical, flag, nominal or ordinal.

Target fields. Only one target field can be specified. For C&R Tree, CHAID, Tree-AS, and Random Trees, the target can be continuous, categorical, flag, nominal or ordinal. For QUEST it can be categorical, flag or nominal. For C5.0 the target can be flag, nominal or ordinal.

Type of split. C&R Tree, QUEST, and Random Trees support only binary splits (that is, each node of the tree can be split into no more than two branches). By contrast, CHAID, C5.0, and Tree-AS support splitting into more than two branches at a time.

Method used for splitting. The algorithms differ in the criteria used to decide the splits. When C&R Tree predicts a categorical output, a dispersion measure is used (by default the Gini coefficient, though you can change this). For continuous targets, the least squared deviation method is used. CHAID and Tree-AS use a chi-square test; QUEST uses a chi-square test for categorical predictors, and analysis of variance for continuous inputs. For C5.0 an information theory measure is used, the information gain ratio.

Missing value handling. All algorithms allow missing values for the predictor fields, though they use different methods to handle them. C&R Tree and QUEST use substitute prediction fields, where needed, to advance a record with missing values through the tree during training. CHAID makes the missing values a separate category and enables them to be used in tree building. C5.0 uses a fractioning method, which passes a fractional part of a record down each branch of the tree from a node where the split is based on a field with a missing value.

Pruning. C&R Tree, QUEST and C5.0 offer the option to grow the tree fully and then prune it back by removing bottom-level splits that do not contribute significantly to the accuracy of the tree. However, all of the decision tree algorithms allow you to control the minimum subgroup size, which helps avoid branches with few data records.

Interactive tree building. C&R Tree, QUEST and CHAID provide an option to launch an interactive session. This enables you to build your tree one level at a time, edit the splits, and prune the tree before you create the model. C5.0, Tree-AS, and Random Trees do not have an interactive option.

Prior probabilities. C&R Tree and QUEST support the specification of prior probabilities for categories when predicting a categorical target field. Prior probabilities are estimates of the overall relative frequency for each target category in the population from which the training data are drawn. In other words, they are the probability estimates that you would make for each possible target value prior to knowing anything about predictor values. CHAID, C5.0, Tree-AS, and Random Trees do not support specifying prior probabilities.

Rule sets. Not available for Tree-AS or Random Trees. For models with categorical target fields, the decision tree nodes provide the option to create the model in the form of a rule set, which can sometimes be easier to interpret than a complex decision tree. For C&R Tree, QUEST and CHAID you can generate a rule set from an interactive session; for C5.0 you can specify this option on the modeling node. In addition, all decision tree models enable you to generate a rule set from the model nugget. See the topic [Generating a Rule Set from a Decision Tree](#) for more information.

- [C&R Tree Node](#)
- [CHAID Node](#)
- [QUEST Node](#)
- [Decision Tree Node Fields Options](#)
- [Decision Tree Node Build Options](#)
- [Decision Tree Node Model Options](#)

C&R Tree Node

The Classification and Regression (C&R) Tree node is a tree-based classification and prediction method. Similar to C5.0, this method uses recursive partitioning to split the training records into segments with similar output field values. The C&R Tree node starts by examining the input fields to find the best split, measured by the reduction in an impurity index that results from the split. The split defines two subgroups, each of which is subsequently split into two more subgroups, and so on, until one of the stopping criteria is triggered. All splits are binary (only two subgroups).

Pruning

C&R Trees give you the option to first grow the tree and then prune based on a cost-complexity algorithm that adjusts the risk estimate based on the number of terminal nodes. This method, which enables the tree to grow large before pruning based on more complex criteria, may result in smaller trees with better cross-validation properties. Increasing the number of terminal nodes generally reduces the risk for the current (training) data, but the actual risk may be higher when the model is generalized to unseen data. In an extreme case, suppose you have a separate terminal node for each record in the training set. The risk estimate would be 0%, since every record falls into its own node, but the risk of misclassification for unseen (testing) data would almost certainly be greater than 0. The cost-complexity measure attempts to compensate for this.

Example. A cable TV company has commissioned a marketing study to determine which customers would buy a subscription to an interactive news service via cable. Using the data from the study, you can create a stream in which the target field is the intent to buy the subscription and the predictor fields include age, sex, education, income category, hours spent watching television each day, and number of children. By applying a C&R Tree node to the stream, you will be able to predict and classify the responses to get the highest response rate for your campaign.

Requirements. To train a C&R Tree model, you need one or more *Input* fields and exactly one *Target* field. Target and input fields can be continuous (numeric range) or categorical. Fields set to *Both* or *None* are ignored. Fields used in the model must have their types fully

instantiated, and any ordinal (ordered set) fields used in the model must have numeric storage (not string). If necessary, the Reclassify node can be used to convert them.

Strengths. C&R Tree models are quite robust in the presence of problems such as missing data and large numbers of fields. They usually do not require long training times to estimate. In addition, C&R Tree models tend to be easier to understand than some other model types--the rules derived from the model have a very straightforward interpretation. Unlike C5.0, C&R Tree can accommodate continuous as well as categorical output fields.

Related information

- [Tree-Growing Directives](#)
-

CHAID Node

CHAID, or Chi-squared Automatic Interaction Detection, is a classification method for building decision trees by using chi-square statistics to identify optimal splits.

CHAID first examines the crosstabulations between each of the input fields and the outcome, and tests for significance using a chi-square independence test. If more than one of these relations is statistically significant, CHAID will select the input field that is the most significant (smallest p value). If an input has more than two categories, these are compared, and categories that show no differences in the outcome are collapsed together. This is done by successively joining the pair of categories showing the least significant difference. This category-merging process stops when all remaining categories differ at the specified testing level. For nominal input fields, any categories can be merged; for an ordinal set, only contiguous categories can be merged.

Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits for each predictor but takes longer to compute.

Requirements. Target and input fields can be continuous or categorical; nodes can be split into two or more subgroups at each level. Any ordinal fields used in the model must have numeric storage (not string). If necessary, the Reclassify node can be used to convert them.

Strengths. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. It therefore tends to create a wider tree than the binary growing methods. CHAID works for all types of inputs, and it accepts both case weights and frequency variables.

QUEST Node

QUEST—or Quick, Unbiased, Efficient Statistical Tree—is a binary classification method for building decision trees. A major motivation in its development was to reduce the processing time required for large C&R Tree analyses with either many variables or many cases. A second goal of QUEST was to reduce the tendency found in classification tree methods to favor inputs that allow more splits, that is, continuous (numeric range) input fields or those with many categories.

- QUEST uses a sequence of rules, based on significance tests, to evaluate the input fields at a node. For selection purposes, as little as a single test may need to be performed on each input at a node. Unlike C&R Tree, all splits are not examined, and unlike C&R Tree and CHAID, category combinations are not tested when evaluating an input field for selection. This speeds the analysis.
- Splits are determined by running quadratic discriminant analysis using the selected input on groups formed by the target categories. This method again results in a speed improvement over exhaustive search (C&R Tree) to determine the optimal split.

Requirements. Input fields can be continuous (numeric ranges), but the target field must be categorical. All splits are binary. Weight fields cannot be used. Any ordinal (ordered set) fields used in the model must have numeric storage (not string). If necessary, the Reclassify node can be used to convert them.

Strengths. Like CHAID, but unlike C&R Tree, QUEST uses statistical tests to decide whether or not an input field is used. It also separates the issues of input selection and splitting, applying different criteria to each. This contrasts with CHAID, in which the statistical test result that determines variable selection also produces the split. Similarly, C&R Tree employs the impurity-change measure to both select the input field and to determine the split.

Related information

- [Tree-Growing Directives](#)
-

Decision Tree Node Fields Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign targets, predictors and other roles, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Target. Select one field as the target for the prediction.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

Analysis Weight. (CHAID, C&RT, and Trees-AS only) To use a field as a case weight, specify the field here. Case weights are used to account for differences in variance across levels of the output field. See the topic [Using Frequency and Weight Fields](#) for more information.

Decision Tree Node Build Options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains several different panes on which you set the customizations that are specific to your model.

- [Decision Tree Nodes - Objectives](#)
- [Decision Tree Nodes - Basics](#)
- [Decision Tree Nodes - Stopping Rules](#)
- [Decision Tree Nodes - Ensembles](#)
- [C&R Tree and QUEST Nodes - Costs & Priors](#)
- [CHAID Node - Costs](#)
- [C&R Tree Node - Advanced](#)
- [QUEST Node - Advanced](#)
- [CHAID Node - Advanced](#)

Decision Tree Nodes - Objectives

For the C&R Tree, QUEST, and CHAID nodes, in the Objectives pane on the Build Options tab, you can choose whether to build a new model or update an existing one. You also set the main objective of the node: to build a standard model, to build one with enhanced accuracy or stability, or to build one for use with very large datasets.

What do you want to do?

Build new model. (Default) Creates a completely new model each time you run a stream containing this modeling node.

Continue training existing model. By default, a completely new model is created each time a modeling node is executed. If this option is selected, training continues with the last model successfully produced by the node. This makes it possible to update or refresh an existing model without having to access the original data and may result in significantly faster performance since *only* the new or updated records are fed into the stream. Details on the previous model are stored with the modeling node, making it possible to use this option even if the previous model nugget is no longer available in the stream or Models palette.

Note: This option is activated only if you select Build a single tree (for C&R Tree, CHAID, and QUEST), Create a standard model (for Neural Net and Linear), or Create a model for very large datasets as the objective.

What is your main objective?

- Build a single tree. Creates a single, standard decision tree model. Standard models are generally easier to interpret, and can be faster to score, than models built using the other objective options.

Note: Continue training existing model is only supported with Build a single tree split models, and you must be connected to Analytic Server.

Mode. Specifies the method used to build the model. Generate model creates a model automatically when the stream is run. Launch interactive session opens the tree builder, which enables you to build your tree one level at a time, edit splits, and prune as desired before

creating the model nugget.

Use tree directives. Select this option to specify directives to apply when generating an interactive tree from the node. For example, you can specify the first- and second-level splits, and these would automatically be applied when the tree builder is launched. You can also save directives from an interactive tree-building session in order to re-create the tree at a future date. See the topic [Updating Tree Directives](#) for more information.

- Enhance model accuracy (boosting). Choose this option if you want to use a special method, known as **boosting**, to improve the model accuracy rate. Boosting works by building multiple models in a sequence. The first model is built in the usual way. Then, a second model is built in such a way that it focuses on the records that were misclassified by the first model. Then a third model is built to focus on the second model's errors, and so on. Finally, cases are classified by applying the whole set of models to them, using a weighted voting procedure to combine the separate predictions into one overall prediction. Boosting can significantly improve the accuracy of a decision tree model, but it also requires longer training.
- Enhance model stability (bagging). Choose this option if you want to use a special method, known as **bagging** (bootstrap aggregating), to improve the stability of the model and to avoid overfitting. This option creates multiple models and combines them, in order to obtain more reliable predictions. Models obtained using this option can take longer to build and score than standard models.
- Create a model for very large datasets. Choose this option when working with datasets that are too large to build a model using any of the other objective options. This option divides the data into smaller data blocks and builds a model on each block. The most accurate models are then automatically selected and combined into a single model nugget. You can perform incremental model updating if you select the Continue training existing model option on this screen. The Continue training existing model option is only supported with Create a model for very large datasets models, and you don't need to connect to Analytic Server. But a model for very large datasets can't be created with splits.

Note: This option for very large datasets requires a connection to IBM® SPSS® Modeler Server.

Decision Tree Nodes - Basics

Specify the basic options about how the decision tree is to be built.

Tree growing algorithm (CHAID and Tree-AS only) Choose the type of CHAID algorithm you want to use. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits for each predictor but takes longer to compute.

Maximum tree depth Specify the maximum number of levels below the root node (the number of times the sample will be split recursively). The default is 5; choose Custom and enter a value to specify a different number of levels.

Pruning (C&RT and QUEST only)

Prune tree to avoid overfitting Pruning consists of removing bottom-level splits that do not contribute significantly to the accuracy of the tree. Pruning can help simplify the tree, making it easier to interpret and, in some cases, improving generalization. If you want the full tree without pruning, leave this option deselected.

- Set maximum difference in risk (in Standard Errors) Enables you to specify a more liberal pruning rule. The standard error rule enables the algorithm to select the simplest tree whose risk estimate is close to (but possibly greater than) that of the subtree with the smallest risk. The value indicates the size of the allowable difference in the risk estimate between the pruned tree and the tree with the smallest risk in terms of the risk estimate. For example, if you specify 2, a tree whose risk estimate is $(2 \times \text{standard error})$ larger than that of the full tree could be selected.

Maximum surrogates. Surrogates are a method for dealing with missing values. For each split in the tree, the algorithm identifies the input fields that are most similar to the selected split field. Those fields are the *surrogates* for that split. When a record must be classified but has a missing value for a split field, its value on a surrogate field can be used to make the split. Increasing this setting will allow more flexibility to handle missing values but may also lead to increased memory usage and longer training times.

Decision Tree Nodes - Stopping Rules

These options control how the tree is constructed. Stopping rules determine when to stop splitting specific branches of the tree. Set the minimum branch sizes to prevent splits that would create very small subgroups. Minimum records in parent branch prevents a split if the number of records in the node to be split (the *parent*) is less than the specified value. Minimum records in child branch prevents a split if the number of records in any branch created by the split (the *child*) would be less than the specified value.

- Use percentage Specify sizes in terms of percentage of overall training data.
- Use absolute value Specify sizes as the absolute numbers of records.

Decision Tree Nodes - Ensembles

These settings determine the behavior of ensembling that occurs when boosting, bagging, or very large datasets are requested in Objectives. Options that do not apply to the selected objective are ignored.

Bagging and Very Large Datasets. When scoring an ensemble, this is the rule used to combine the predicted values from the base models to compute the ensemble score value.

- Default combining rule for categorical targets. Ensemble predicted values for categorical targets can be combined using voting, highest probability, or highest mean probability. **Voting** selects the category that has the highest probability most often across the base models. **Highest probability** selects the category that achieves the single highest probability across all base models. **Highest mean probability** selects the category with the highest value when the category probabilities are averaged across base models.
- Default combining rule for continuous targets. Ensemble predicted values for continuous targets can be combined using the mean or median of the predicted values from the base models.

Note that when the objective is to enhance model accuracy, the combining rule selections are ignored. Boosting always uses a weighted majority vote to score categorical targets and a weighted median to score continuous targets.

Boosting and Bagging. Specify the number of base models to build when the objective is to enhance model accuracy or stability; for bagging, this is the number of bootstrap samples. It should be a positive integer.

C&R Tree and QUEST Nodes - Costs & Priors

Misclassification Costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

With the exception of C5.0 models, misclassification costs are not applied when scoring a model and are not taken into account when ranking or comparing models using an Auto Classifier node, evaluation chart, or Analysis node. A model that includes costs may not produce fewer errors than one that doesn't and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of *less expensive* errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select Use misclassification costs and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying A as B to be 2.0, the cost of misclassifying B as A will still have the default value of 1.0 unless you explicitly change it as well.

Priors

These options allow you to specify prior probabilities for categories when predicting a categorical target field. **Prior probabilities** are estimates of the overall relative frequency for each target category in the population from which the training data are drawn. In other words, they are the probability estimates that you would make for each possible target value *prior* to knowing anything about predictor values. There are three methods of setting priors:

- Based on training data. This is the default. Prior probabilities are based on the relative frequencies of the categories in the training data.
- Equal for all classes. Prior probabilities for all categories are defined as $1/k$, where k is the number of target categories.
- Custom. You can specify your own prior probabilities. Starting values for prior probabilities are set as equal for all classes. You can adjust the probabilities for individual categories to user-defined values. To adjust a specific category's probability, select the probability cell in the table corresponding to the desired category, delete the contents of the cell, and enter the desired value.

The prior probabilities for all categories should sum to 1.0 (the **probability constraint**). If they do not sum to 1.0, a warning is displayed, with an option to automatically normalize the values. This automatic adjustment preserves the proportions across categories while enforcing the probability constraint. You can perform this adjustment at any time by clicking the Normalize button. To reset the table to equal values for all categories, click the Equalize button.

Adjust priors using misclassification costs. This option enables you to adjust the priors, based on misclassification costs (specified on the Costs tab). This enables you to incorporate cost information directly into the tree-growing process for trees that use the Twoing impurity measure. (When this option is not selected, cost information is used only in classifying records and calculating risk estimates for trees based on the Twoing measure.)

CHAID Node - Costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

With the exception of C5.0 models, misclassification costs are not applied when scoring a model and are not taken into account when ranking or comparing models using an Auto Classifier node, evaluation chart, or Analysis node. A model that includes costs may not produce fewer errors than one that doesn't and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of *less expensive* errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select Use misclassification costs and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying A as B to be 2.0, the cost of misclassifying B as A will still have the default value of 1.0 unless you explicitly change it as well.

C&R Tree Node - Advanced

The advanced options enable you to fine-tune the tree-building process.

Minimum change in impurity. Specify the minimum change in impurity to create a new split in the tree. **Impurity** refers to the extent to which subgroups defined by the tree have a wide range of output field values within each group. For categorical targets, a node is considered “pure” if 100% of cases in the node fall into a specific category of the target field. The goal of tree building is to create subgroups with similar output values--in other words, to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the specified amount, the split will not be made.

Impurity measure for categorical targets. For categorical target fields, specify the method used to measure the impurity of the tree. (For continuous targets, this option is ignored, and the **least squared deviation** impurity measure is always used.)

- Gini is a general impurity measure based on probabilities of category membership for the branch.
- Twoing is an impurity measure that emphasizes the binary split and is more likely to lead to approximately equal-sized branches from a split.
- Ordered adds the additional constraint that only contiguous target classes can be grouped together, as is applicable only with ordinal targets. If this option is selected for a nominal target, the standard twoing measure is used by default.

Overfit prevention set. The algorithm internally separates records into a model building set and an overfit prevention set, which is an independent set of data records used to track errors during training in order to prevent the method from modeling chance variation in the data. Specify a percentage of records. The default is 30.

Replicate results. Setting a random seed enables you to replicate analyses. Specify an integer or click Generate, which will create a pseudo-random integer between 1 and 2147483647, inclusive.

QUEST Node - Advanced

The advanced options enable you to fine-tune the tree-building process.

Significance level for splitting. Specifies the significance level (alpha) for splitting nodes. The value must be between 0 and 1. Lower values tend to produce trees with fewer nodes.

Overfit prevention set. The algorithm internally separates records into a model building set and an overfit prevention set, which is an independent set of data records used to track errors during training in order to prevent the method from modeling chance variation in the data. Specify a percentage of records. The default is 30.

Replicate results. Setting a random seed enables you to replicate analyses. Specify an integer or click Generate, which will create a pseudo-random integer between 1 and 2147483647, inclusive.

CHAID Node - Advanced

The advanced options enable you to fine-tune the tree-building process.

Significance level for splitting. Specifies the significance level (alpha) for splitting nodes. The value must be between 0 and 1. Lower values tend to produce trees with fewer nodes.

Significance level for merging. Specifies the significance level (alpha) for merging categories. The value must be greater than 0 and less than or equal to 1. To prevent any merging of categories, specify a value of 1. For continuous targets, this means the number of categories for the variable in the final tree matches the specified number of intervals. This option is not available for Exhaustive CHAID.

Adjust significance values using Bonferroni method. Adjusts significance values when testing the various category combinations of a predictor. Values are adjusted based on the number of tests, which directly relates to the number of categories and measurement level of a predictor. This is generally desirable because it better controls the false-positive error rate. Disabling this option will increase the power of your analysis to find true differences, but at the cost of an increased false-positive rate. In particular, disabling this option may be recommended for small samples.

Allow resplitting of merged categories within a node. The CHAID algorithm attempts to merge categories in order to produce the simplest tree that describes the model. If selected, this option enables merged categories to be resplit if that results in a better solution.

Chi-square for categorical targets. For categorical targets, you can specify the method used to calculate the chi-square statistic.

- **Pearson.** This method provides faster calculations but should be used with caution on small samples.
- **Likelihood ratio.** This method is more robust than Pearson but takes longer to calculate. For small samples, this is the preferred method. For continuous targets, this method is always used.

Minimum change in expected cell frequencies. When estimating cell frequencies (for both the nominal model and the row effects ordinal model), an iterative procedure (epsilon) is used to converge on the optimal estimate used in the chi-square test for a specific split. Epsilon determines how much change must occur for iterations to continue; if the change from the last iteration is smaller than the specified value, iterations stop. If you are having problems with the algorithm not converging, you can increase this value or increase the maximum number of iterations until convergence occurs.

Maximum iterations for convergence. Specifies the maximum number of iterations before stopping, whether convergence has taken place or not.

Overfit prevention set. (This option is only available when using the interactive tree builder.) The algorithm internally separates records into a model building set and an overfit prevention set, which is an independent set of data records used to track errors during training in order to prevent the method from modeling chance variation in the data. Specify a percentage of records. The default is 30.

Replicate results. Setting a random seed enables you to replicate analyses. Specify an integer or click Generate, which will create a pseudo-random integer between 1 and 2147483647, inclusive.

Decision Tree Node Model Options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also choose to obtain predictor importance information, as well as raw and adjusted propensity scores for flag targets.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Model Evaluation

Calculate predictor importance. For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may take longer to calculate for some models, particularly when working with large datasets, and is off by default for some models as a result. Predictor importance is not available for decision list models. See [Predictor Importance](#) for more information.

Propensity Scores

Propensity scores can be enabled in the modeling node, and on the Settings tab in the model nugget. This functionality is available only when the selected target is a flag field. See the topic [Propensity Scores](#) for more information.

Calculate raw propensity scores. Raw propensity scores are derived from the model based on the training data only. If the model predicts the true value (will respond), then the propensity is the same as P, where P is the probability of the prediction. If the model predicts the false value, then the propensity is calculated as $(1 - P)$.

- If you choose this option when building the model, propensity scores will be enabled in the model nugget by default. However, you can always choose to enable raw propensity scores in the model nugget whether or not you select them in the modeling node.
- When scoring the model, raw propensity scores will be added in a field with the letters *RP* appended to the standard prefix. For example, if the predictions are in a field named *\$R-churn*, the name of the propensity score field will be *\$RRP-churn*.

Calculate adjusted propensity scores. Raw propensities are based purely on estimates given by the model, which may be overfitted, leading to over-optimistic estimates of propensity. Adjusted propensities attempt to compensate by looking at how the model performs on the test or validation partitions and adjusting the propensities to give a better estimate accordingly.

- This setting requires that a valid partition field is present in the stream.
- Unlike raw confidence scores, adjusted propensity scores must be calculated when building the model; otherwise, they will not be available when scoring the model nugget.
- When scoring the model, adjusted propensity scores will be added in a field with the letters *AP* appended to the standard prefix. For example, if the predictions are in a field named *\$R-churn*, the name of the propensity score field will be *\$RAP-churn*. Adjusted propensity scores are not available for logistic regression models.
- When calculating the adjusted propensity scores, the test or validation partition used for the calculation must not have been balanced. To avoid this, be sure the Only balance training data option is selected in any upstream Balance nodes. In addition, if a complex sample has been taken upstream this will invalidate the adjusted propensity scores.
- Adjusted propensity scores are not available for "boosted" tree and rule set models. See the topic [Boosted C5.0 Models](#) for more information.

Based on. For adjusted propensity scores to be computed, a partition field must be present in the stream. You can specify whether to use the testing or validation partition for this computation. For best results, the testing or validation partition should include at least as many records as the partition used to train the original model.

C5.0 Node

This feature is available in SPSS® Modeler Professional and SPSS Modeler Premium.

This node uses the C5.0 algorithm to build either a **decision tree** or a **rule set**. A C5.0 model works by splitting the sample based on the field that provides the maximum **information gain**. Each subsample defined by the first split is then split again, usually based on a different field, and the process repeats until the subsamples cannot be split any further. Finally, the lowest-level splits are reexamined, and those that do not contribute significantly to the value of the model are removed or **pruned**.

Note: The C5.0 node can predict only a categorical target. When analyzing data with categorical (nominal or ordinal) fields, the node is more likely to group categories together than versions of C5.0 prior to release 11.0.

C5.0 can produce two kinds of models. A **decision tree** is a straightforward description of the splits found by the algorithm. Each terminal (or "leaf") node describes a particular subset of the training data, and each case in the training data belongs to exactly one terminal node in the tree. In other words, exactly one prediction is possible for any particular data record presented to a decision tree.

In contrast, a **rule set** is a set of rules that tries to make predictions for individual records. Rule sets are derived from decision trees and, in a way, represent a simplified or distilled version of the information found in the decision tree. Rule sets can often retain most of the important information from a full decision tree but with a less complex model. Because of the way rule sets work, they do not have the same properties as decision trees. The most important difference is that with a rule set, more than one rule may apply for any particular record, or no rules at all may apply. If multiple rules apply, each rule gets a weighted "vote" based on the confidence associated with that rule, and the final prediction is decided by combining the weighted votes of all of the rules that apply to the record in question. If no rule applies, a default prediction is assigned to the record.

Example. A medical researcher has collected data about a set of patients, all of whom suffered from the same illness. During their course of treatment, each patient responded to one of five medications. You can use a C5.0 model, in conjunction with other nodes, to help find out which drug might be appropriate for a future patient with the same illness.

Requirements. To train a C5.0 model, there must be one categorical (i.e., nominal or ordinal) *Target* field, and one or more *Input* fields of any type. Fields set to *Both* or *None* are ignored. Fields used in the model must have their types fully instantiated. A weight field can also be specified.

Strengths. C5.0 models are quite robust in the presence of problems such as missing data and large numbers of input fields. They usually do not require long training times to estimate. In addition, C5.0 models tend to be easier to understand than some other model types, since the rules derived from the model have a very straightforward interpretation. C5.0 also offers the powerful **boosting** method to increase accuracy of classification.

Note: C5.0 model building speed may benefit from enabling parallel processing.

- [C5.0 Node Model Options](#)

Related information

- [C5.0 Node Model Options](#)

C5.0 Node Model Options

This feature is available in SPSS® Modeler Professional and SPSS Modeler Premium.

Model name. Specify the name of the model to be produced.

- **Auto.** With this option selected, the model name will be generated automatically, based on the target field name(s). This is the default.
- **Custom.** Select this option to specify your own name for the model nugget that will be created by this node.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Create split models. Builds a separate model for each possible value of input fields that are specified as split fields. See [Building Split Models](#) for more information.

Output type. Specify here whether you want the resulting model nugget to be a Decision tree or a Rule set.

Group symbolics. If this option is selected, C5.0 will attempt to combine symbolic values that have similar patterns with respect to the output field. If this option is not selected, C5.0 will create a child node for every value of the symbolic field used to split the parent node. For example, if C5.0 splits on a *COLOR* field (with values *RED*, *GREEN*, and *BLUE*), it will create a three-way split by default. However, if this option is selected, and the records where *COLOR = RED* are very similar to records where *COLOR = BLUE*, it will create a two-way split, with the *GREENs* in one group and the *BLUES* and *REDS* together in the other.

Use boosting. The C5.0 algorithm has a special method for improving its accuracy rate, called **boosting**. It works by building multiple models in a sequence. The first model is built in the usual way. Then, a second model is built in such a way that it focuses on the records that were misclassified by the first model. Then a third model is built to focus on the second model's errors, and so on. Finally, cases are classified by applying the whole set of models to them, using a weighted voting procedure to combine the separate predictions into one overall prediction. Boosting can significantly improve the accuracy of a C5.0 model, but it also requires longer training. The Number of trials option enables you to control how many models are used for the boosted model.

Cross-validate. If this option is selected, C5.0 will use a set of models built on subsets of the training data to estimate the accuracy of a model built on the full dataset. This is useful if your dataset is too small to split into traditional training and testing sets. The cross-validation models are discarded after the accuracy estimate is calculated. You can specify the **number of folds**, or the number of models used for cross-validation. Note that in previous versions of IBM® SPSS Modeler, building the model and cross-validating it were two separate operations. In the current version, no separate model-building step is required. Model building and cross-validation are performed at the same time.

Mode. For Simple training, most of the C5.0 parameters are set automatically. Expert training allows more direct control over the training parameters.

Simple Mode Options

Favor. By default, C5.0 will try to produce the most accurate tree possible. In some instances, this can lead to overfitting, which can result in poor performance when the model is applied to new data. Select Generality to use algorithm settings that are less susceptible to this problem.

Note: Models built with the Generality option selected are not guaranteed to generalize better than other models. When generality is a critical issue, always validate your model against a held-out test sample.

Expected noise (%). Specify the expected proportion of noisy or erroneous data in the training set.

Expert Mode Options

Pruning severity. Determines the extent to which the decision tree or rule set will be pruned. Increase this value to obtain a smaller, more concise tree. Decrease it to obtain a more accurate tree. This setting affects local pruning only (see "Use global pruning" below).

Minimum records per child branch. The size of subgroups can be used to limit the number of splits in any branch of the tree. A branch of the tree will be split only if two or more of the resulting subbranches would contain at least this many records from the training set. The default value is 2. Increase this value to help prevent **overtraining** with noisy data.

Use global pruning. Trees are pruned in two stages: First, a local pruning stage, which examines subtrees and collapses branches to increase the accuracy of the model. Second, a global pruning stage considers the tree as a whole, and weak subtrees may be collapsed. Global pruning is performed by default. To omit the global pruning stage, deselect this option.

Winnow attributes. If this option is selected, C5.0 will examine the usefulness of the predictors before starting to build the model. Predictors that are found to be irrelevant are then excluded from the model-building process. This option can be helpful for models with many predictor fields and can help prevent overfitting.

Note: C5.0 model building speed may benefit from enabling parallel processing.

Related information

- [C5.0 Node](#)

Tree-AS node

The Tree-AS node can be used with data in a distributed environment. In this node you can choose to build decision trees using either a CHAID or Exhaustive CHAID model.

CHAID, or Chi-squared Automatic Interaction Detection, is a classification method for building decision trees by using chi-square statistics to identify optimal splits.

CHAID first examines the crosstabulations between each of the input fields and the outcome, and tests for significance using a chi-square independence test. If more than one of these relations is statistically significant, CHAID will select the input field that is the most significant (smallest p value). If an input has more than two categories, these are compared, and categories that show no differences in the outcome are collapsed together. This is done by successively joining the pair of categories showing the least significant difference. This category-merging process stops when all remaining categories differ at the specified testing level. For nominal input fields, any categories can be merged; for an ordinal set, only contiguous categories can be merged.

Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits for each predictor but takes longer to compute.

Requirements. Target and input fields can be continuous or categorical; nodes can be split into two or more subgroups at each level. Any ordinal fields used in the model must have numeric storage (not string). If necessary, use the Reclassify node to convert them.

Strengths. CHAID can generate nonbinary trees, meaning that some splits have more than two branches. It therefore tends to create a wider tree than the binary growing methods. CHAID works for all types of inputs, and it accepts both case weights and frequency variables.

- [Tree-AS node fields options](#)
 - [Tree-AS node build options](#)
 - [Tree-AS node model options](#)
 - [Tree-AS model nugget](#)
-

Tree-AS node fields options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign targets, predictors and other roles, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Target. Select one field as the target for the prediction.

Predictors Select one or more fields as inputs for the prediction.

Analysis Weight To use a field as a case weight, specify the field here. Case weights are used to account for differences in variance across levels of the output field. For more information, see [Using Frequency and Weight Fields](#).

Tree-AS node build options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains several different panes on which you set the customizations that are specific to your model.

- [Tree-AS node - basics](#)
 - [Tree-AS node - growing](#)
 - [Tree-AS node - stopping rules](#)
 - [Tree-AS node - costs](#)
-

Tree-AS node - basics

Specify the basic options about how the decision tree is to be built.

Tree growing algorithm Select the type of CHAID algorithm you want to use. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits for each predictor but takes longer to compute.

Maximum tree depth Specify the maximum number of levels below the root node (the number of times the sample is split recursively); the default is 5. The maximum number of levels (also referred to as *nodes*) is 50,000.

Binning If you use continuous data, you must bin the inputs. You can do this in a preceding node; however, the Tree-AS node automatically bins any continuous inputs. If you use the Tree-AS node to automatically bin the data, select the Number of bins into which the inputs are to be divided. Data is divided into bins with equal frequency; the available options are 2, 4, 5, 10, 20, 25, 50, or 100.

Tree-AS node - growing

Use the growing options to fine-tune the tree-building process.

Record threshold for switching from p-values to effect sizes Specify the number of records at which the model will switch from using the P-values settings to the Effect size settings when building the tree. The default is 1,000,000.

Significance level for splitting Specify the significance level (alpha) for splitting nodes. The value must be between 0.01 and 0.99. Lower values tend to produce trees with fewer nodes.

Significance level for merging Specify the significance level (alpha) for merging categories. The value must be between 0.01 and 0.99. This option is not available for Exhaustive CHAID.

Adjust significance values using Bonferroni method Adjust significance values when you are testing the various category combinations of a predictor. Values are adjusted based on the number of tests, which directly relates to the number of categories and measurement level of a predictor. This is generally desirable because it better controls the false-positive error rate. Disabling this option increases the power of your analysis to find true differences, but at the cost of an increased false-positive rate. In particular, disabling this option may be recommended for small samples.

Effect size threshold (continuous targets only) Set the effect size threshold to be used when splitting nodes and merging categories; when using a continuous target. The value must be between 0.01 and 0.99.

Effect size threshold (categorical targets only) Set the effect size threshold to be used when splitting nodes and merging categories; when using a categorical target. The value must be between 0.01 and 0.99.

Allow resplitting of merged categories within a node The CHAID algorithm attempts to merge categories in order to produce the simplest tree that describes the model. If selected, this option enables merged categories to be resplit if that results in a better solution.

Significance level for grouping leaf nodes Specify the significance level that determines how groups of leaf nodes are formed or how unusual leaf nodes are identified.

Chi-square for categorical targets For categorical targets, you can specify the method used to calculate the chi-square statistic.

- Pearson This method provides faster calculations but should be used with caution on small samples.
- Likelihood ratio This method is more robust than Pearson but takes longer to calculate. For small samples, this is the preferred method. For continuous targets, this method is always used.

Tree-AS node - stopping rules

These options control how the tree is constructed. Stopping rules determine when to stop splitting specific branches of the tree. Set the minimum branch sizes to prevent splits that would create very small subgroups. Minimum records in parent branch prevents a split if the number of records in the node to be split (the *parent*) is less than the specified value. Minimum records in child branch prevents a split if the number of records in any branch created by the split (the *child*) would be less than the specified value.

- Use percentage Specify sizes in terms of percentage of overall training data.
- Use absolute value Specify sizes as the absolute numbers of records.

Minimum change in expected cell frequencies When estimating cell frequencies (for both the nominal model and the row effects ordinal model), an iterative procedure (epsilon) is used to converge on the optimal estimate used in the chi-square test for a specific split. Epsilon determines how much change must occur for iterations to continue; if the change from the last iteration is smaller than the specified value, iterations stop. If you are having problems with the algorithm not converging, you can increase this value or increase the maximum number of iterations until convergence occurs.

Maximum iterations for convergence Specifies the maximum number of iterations before stopping, whether convergence has taken place or not.

Tree-AS node - costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

A model that includes costs might not produce fewer errors than one that doesn't, and might not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of less expensive errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select Use misclassification costs and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying A as B to be 2.0, the cost of misclassifying B as A will still have the default value of 1.0 unless you explicitly change it as well.

For ordinal targets only, you can select the Default cost increase for ordinal target and set default values in the costs matrix. The available options are described in the following list.

- No increase - A default value of 1.0 for every correct prediction.
- Linear - Each successive incorrect prediction increases the cost by 1.
- Square - Each successive incorrect prediction is the square of the linear value. In this case, the values might be: 1, 4, 9, and so on.
- Custom - If you manually edit any values in the table, the drop-down option automatically changes to Custom. If you change the drop-down selection to any of the other options your edited values are replaced with the values for the selected option.

Tree-AS node model options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also choose to calculate confidence values and add an identifying ID during scoring of the model.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Calculate confidences To add a confidence field when the model is scored, select this check box.

Rule identifier To add a field when the model is scored that contains the ID of the leaf node a record was assigned to, select this check box.

Tree-AS model nugget

- [Tree-AS model nugget output](#)
- [Tree-AS model nugget settings](#)

Tree-AS model nugget output

After you create a Tree-AS model, the following information is available in the Output viewer.

Model Information table

The Model Information table provides key information about the model. The table identifies some high-level model settings, such as:

- The algorithm type used; either CHAID or Exhaustive CHAID.
- The name of the target field selected in either the Type node or the Tree-AS node Fields tab.
- The names of the fields selected as predictors in either the Type node or the Tree-AS node Fields tab.
- The number of records in the data. If you build a model with a frequency weight, this value is the weighted, valid count which represents the records on which the tree is based.
- The number of *leaf nodes* in the generated tree.
- The number of levels in the tree; that is, the tree depth.

Predictor Importance

The Predictor Importance graph shows the importance of the top 10 inputs (predictors) in the model as a bar chart.

If there are more than 10 fields in the chart, you can change the selection of predictors that are included in the chart by using the slider beneath the chart. The indicator marks on the slider are a fixed width, and each mark on the slider represents 10 fields. You can move the indicator marks along the slider to display the next or previous 10 fields, ordered by predictor importance.

You can double-click the chart to open a separate dialog box in which you can edit the graph settings. For example, you can amend items such as the size of the graph, and the size and color of the fonts used. When you close this separate editing dialog box, the changes are applied to the chart that is displayed in the Output tab.

Top Decision Rules table

By default, this interactive table displays the statistics of the rules for the top five leaf nodes in the output, based on the percentage of total records that are contained within the leaf node.

You can double-click the table to open a separate dialog box in which you can edit the rule information that is shown in the table. The information that is displayed, and the options that are available in the dialog box, depend on the data type of the target; such as, categorical, or continuous.

The following rule information is shown in the table:

- Rule ID
- The details of how the rule is applied and made up
- Record count for each rule. If you build a model with a frequency weight, this value is the weighted, valid count which represents the records on which the tree is based.
- Record percentage for each rule

In addition, for a continuous target, an extra column in the table shows the Mean value for each rule.

You can alter the rule table layout using the following Table contents options:

- Top decision rules The top five decision rules are sorted by the percentage of total records contained within the leaf nodes.
- All rules The table contains all of the leaf nodes produced by the model but only shows 20 rules per page. When you select this layout, you can search for a rule by using the additional options of Find rule by ID and Page.

In addition, for a categorical target, you can alter the rule table layout using the Top rules by category option. The top five decision rules are sorted by the percentage of total records for a Target category that you select.

If you change the layout of the rules table, you can copy the modified rules table back to the Output viewer by clicking the Copy to Viewer button at the upper-left of the dialog box.

Tree-AS model nugget settings

On the Settings tab for a Tree-AS model nugget, you specify options for confidences and for SQL generation during model scoring. This tab is available only after the model nugget is added to a stream.

Calculate confidences To include confidences in scoring operations, select this check box. When you score models in the database, excluding confidences means that you can generate more efficient SQL. For regression trees, confidences are not assigned.

Rule identifier To add a field in the scoring output that indicates the ID for the terminal node to which each record is assigned, select this check box.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL is generated:

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL by using the scoring adapter and associated user-defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Random Trees node

The Random Trees node can be used with data in a distributed environment. In this node, you build an ensemble model that consists of multiple decision trees.

The Random Trees node is a tree-based classification and prediction method that is built on Classification and Regression Tree methodology. As with C&R Tree, this prediction method uses recursive partitioning to split the training records into segments with similar output field values. The node starts by examining the input fields available to it to find the best split, which is measured by the reduction in an impurity index that results from the split. The split defines two subgroups, each of which is then split into two more subgroups, and so on, until one of the stopping criteria is triggered. All splits are binary (only two subgroups).

The Random Trees node uses bootstrap sampling with replacement to generate sample data. The sample data is used to grow a tree model. During tree growth, Random Trees will not sample the data again. Instead, it randomly selects part of the predictors and uses the best one to split a tree node. This process is repeated when splitting each tree node. This is the basic idea of growing a tree in random forest.

Random Trees uses C&R Tree-like trees. Since such trees are binary, each field for splitting results in two branches. For a categorical field with multiple categories, the categories are grouped into two groups based on the inner splitting criterion. Each tree grows to the largest extent possible (there is no pruning). In scoring, Random Trees combines individual tree scores by majority voting (for classification) or average (for regression).

Random Trees differ from C&R Trees as follows:

- Random Trees nodes randomly select a specified number of predictors and uses the best one from the selection to split a node. In contrast, C&R Tree finds the best one from all predictors.
- Each tree in Random Trees grows fully until each leaf node typically contains a single record. So the tree depth could be very large. But standard C&R Tree uses different stopping rules for tree growth, which usually leads to a much shallower tree.

Random Trees adds two features compared to C&R Tree:

- The first feature is *bagging*, where replicas of the training dataset are created by sampling with replacement from the original dataset. This action creates bootstrap samples that are of equal size to the original dataset, after which a *component model* is built on each replica. Together these component models form an ensemble model.
- The second feature is that, at each split of the tree, only a sampling of the input fields is considered for the impurity measure.

Requirements. To train a Random Trees model, you need one or more *Input* fields and one *Target* field. Target and input fields can be continuous (numeric range) or categorical. Fields that are set to either *Both* or *None* are ignored. Fields that are used in the model must have their types fully instantiated, and any ordinal (ordered set) fields that are used in the model must have numeric storage (not string). If necessary, the Reclassify node can be used to convert them.

Strengths. Random Trees models are robust when you are dealing with large data sets and numbers of fields. Due to the use of bagging and field sampling, they are much less prone to overfitting and thus the results that are seen in testing are more likely to be repeated when you use new data.

- [Random Trees node fields options](#)
- [Random Trees node build options](#)
- [Random Trees node model options](#)
- [Random Trees model nugget](#)

Random Trees node fields options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign targets, predictors and other roles, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Target. Select one field as the target for the prediction.

Predictors Select one or more fields as inputs for the prediction.

Analysis Weight To use a field as a case weight, specify the field here. Case weights are used to account for differences in variance across levels of the output field. For more information, see [Using Frequency and Weight Fields](#).

Random Trees node build options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains several different panes on which you set the customizations that are specific to your model.

- [Random Trees node - basics](#)
- [Random Trees node - costs](#)
- [Random Trees node - advanced](#)

Random Trees node - basics

Specify basic options for how to build the decision tree.

Number of models to build. Specify the maximum number of trees that the node can build.

Sample size. By default, the size of the bootstrap sample is equal to the original training data. When dealing with large datasets, reducing the sample size can increase performance. It is a ratio from 0 to 1. For example, set the sample size to **0 . 6** to reduce it to 60% of the original training data size.

Handle imbalanced data. If the model's target is a flag outcome (for example, purchase or do not purchase) and the ratio of the desired outcome to non-desired is very small, the data is imbalanced and the bootstrap sampling that is conducted by the model could affect model accuracy. To improve accuracy select this check box; the model then captures a larger proportion of the desired outcome and generates a better model.

Use weighted sampling for variable selection. By default, variables for each leaf node are randomly selected with the same probability. To apply weighting to variables and improve the selection process, select this check box. The weight is calculated by the Random Trees node itself. The more important fields (with higher weight) are more likely to be selected as predictors.

Maximum number of nodes. Specify the maximum number of leaf nodes that are allowed in individual trees. If the number would be exceeded on the next split, tree growth is stopped before the split occurs.

Maximum tree depth. Specify the maximum number of levels *leaf nodes* below the root node; that is, the number of times the sample is split recursively).

Minimum child node size. Specify the minimum number of records that must be contained in a child node after the parent node is split. If a child node would contain fewer records than you enter, the parent node will not be split.

Specify number of predictors to use for splitting. If you are building split models, set the minimum number of predictors to be used to build each split. This prevents the split from creating excessively small subgroups. If you don't select this option, the default value is **[sqrt(M)]** for classification and **[M/3]** for regression, where **M** is the total number of predictor variables. If this option is selected, the specified number of predictors will be used.

Note: The number of predictors for splitting cannot be greater than the total number of predictors in the data.

Stop building when accuracy can no longer be improved. Random Trees uses a particular procedure for deciding when to stop training.

Specifically, if the improvement of the current ensemble accuracy is smaller than a specified threshold, then it will stop adding new trees. This could result in a model with fewer trees than the value you specified for the Number of models to build option.

Random Trees node - costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

A model that includes costs might not produce fewer errors than one that doesn't, and might not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of less expensive errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select Use misclassification costs and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of

misclassifying *A* as *B* to be 2.0, the cost of misclassifying *B* as *A* will still have the default value of 1.0 unless you explicitly change it as well.

For ordinal targets only, you can select the Default cost increase for ordinal target and set default values in the costs matrix. The available options are described in the following list.

- No increase - A default value of 1.0 for every incorrect prediction.
 - Linear - Each successive incorrect prediction increases the cost by 1.
 - Square - Each successive incorrect prediction is the square of the linear value. In this case, the values might be: 1, 4, 9, and so on.
 - Custom - If you manually edit any values in the table, the drop-down option automatically changes to Custom. If you change the drop-down selection to any of the other options your edited values are replaced with the values for the selected option.
-

Random Trees node - advanced

Specify advanced options for how to build the decision tree.

Maximum percentage of missing values. Specify the maximum percentage of missing values allowed in any input. If the percentage exceeds this number, the input is excluded from model building.

Exclude fields with a single category majority over. Specify the maximum percentage of records that a single category can have within a field. If any category value represents a higher percentage of records than specified, the entire field is excluded from model building.

Maximum number of field categories. Specify the maximum number of categories that can be contained within a field. If the number of categories exceeds this number, the field is excluded from model building.

Minimum field variation. If the coefficient of variation of a continuous field is smaller than the value you specify here (in other words, the field is nearly constant), then field is excluded from model building.

Number of bins. Specify the number of equal frequency bins to be used for continuous inputs. The available options are: 2, 4, 5, 10, 20, 25, 50, or 100.

Number of interesting rules to report. Specify the number of rules to report (minimum of 1, maximum of 1000, with a default of 50).

Random Trees node model options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also choose to calculate the importance of the predictors during scoring of the model.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Random Trees model nugget

- [Random Trees model nugget output](#)
 - [Random Trees model nugget settings](#)
-

Random Trees model nugget output

After you create a Random Trees model, the following information is available in the Output viewer:

Model information table

The Model information table provides key information about the model. The table always includes the following high-level model settings:

- The name of the target field that is selected in either the Type node or the Random Trees node Fields tab.
- The model building method - Random Trees.
- The number of predictors input into the model.

The additional details that are shown in the table depend on whether you build a classification or regression model, and if the model is built to handle imbalanced data:

- Classification model (default settings)
 - Model accuracy
 - Misclassification rule
- Classification model (Handle imbalanced data selected)
 - Gmean
 - True positive rate, which is subdivided into classes.
- Regression model
 - Root mean squared error
 - Relative error
 - Variance explained

Records Summary

The summary shows how many records were used to fit the model, and how many were excluded. Both the number of records and the percentage of the whole number are shown. If the model was built to include frequency weight, the unweighted number of records that are included and excluded is also shown.

Predictor Importance

The Predictor Importance graph shows the importance of the top 10 inputs (predictors) in the model as a bar chart.

If there are more than 10 fields in the chart, you can change the selection of predictors that are included in the chart by using the slider beneath the chart. The indicator marks on the slider are a fixed width, and each mark on the slider represents 10 fields. You can move the indicator marks along the slider to display the next or previous 10 fields, ordered by predictor importance.

You can double-click the chart to open a separate dialog box in which you can edit the graph size. When you close this separate editing dialog box, the changes are applied to the chart that is displayed in the Output tab.

Top Decision Rules table

By default, this interactive table displays the statistics of the top rules, which are sorted by interestingness.

You can double-click the table to open a separate dialog box in which you can edit the rule information that is shown in the table. The information that is displayed, and the options that are available in the dialog box, depend on the data type of the target; such as, categorical, or continuous.

The following rule information is shown in the table:

- The details of how the rule is applied and made up
- If the results are in the most frequent category
- Rule accuracy
- Trees accuracy
- Interestingness index

The interestingness index is calculated by using the following formula:

$$I_{Index}(t) = P(A(t)) * P(B(t)) * (P(B(t)|A(t)) + P(\bar{B}(t)|\bar{A}(t)))$$

In this formula:

- $P(A(t))$ is the trees accuracy
- $P(B(t))$ is the rule accuracy
- $P(B(t)|A(t))$ represents correct predictions by both the trees and the node
- The remaining piece of the formula represents incorrect predictions by both the trees and the node.

You can alter the rule table layout by using the following Table contents options:

- Top decision rules The top five decision rules, which are sorted by the interestingness index.
- All rules The table contains all of the rules that are produced by the model but shows only 20 rules per page. When you select this layout, you can search for a rule by using the additional options of Find rule by ID and Page.

In addition, for a categorical target, you can alter the rule table layout by using the Top rules by category option. The top five decision rules are sorted by the percentage of total records for a Target category that you select.

Note: For categorical targets, the table is only available when Handle imbalanced data is not selected in the Basics tab of the Build Options. If you change the layout of the rules table, you can copy the modified rules table back to the Output viewer by clicking the Copy to Viewer button at the upper left of the dialog box.

Confusion Matrix

For classification models, the confusion matrix shows the number of predicted results versus the actual observed results, including the proportion of correct predictions.

Note: The confusion matrix is not available for regression models, nor when Handle imbalanced data is selected in the Basics tab of the Build Options.

Random Trees model nugget settings

On the Settings tab for a Random Trees model nugget, you specify options for confidences and for SQL generation during model scoring. This tab is available only after the model nugget is added to a stream.

Calculate confidences To include confidences in scoring operations, select this check box. When you score models in the database, excluding confidences means that you can generate more efficient SQL. For regression trees, confidences are not assigned.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL is generated:

- **Default:** Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL by using the scoring adapter and associated user-defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

C&R Tree, CHAID, QUEST, and C5.0 decision tree model nuggets

Decision tree model nuggets represent the tree structures for predicting a particular output field discovered by one of the decision tree modeling nodes (C&R Tree, CHAID, QUEST or C5.0). Tree models can be generated directly from the tree-building node, or indirectly from the interactive tree builder. See the topic [The Interactive Tree Builder](#) for more information.

In the model nugget, different options are available depending on the objective specified on the modeling node:

- [single tree model nuggets](#)
- [model nuggets for boosting, bagging and very large datasets](#)

Scoring Tree Models

When you run a stream containing a tree model nugget, the specific result depends on the type of tree.

- For classification trees (categorical target), two new fields, containing the predicted value and the confidence for each record, are added to the data. The prediction is based on the most frequent category for the terminal node to which the record is assigned; if a majority of respondents in a given node is yes, the prediction for all records assigned to that node is yes.
- For regression trees, only predicted values are generated; confidences are not assigned.
- Optionally, for CHAID, QUEST, and C&R Tree models, an additional field can be added that indicates the ID for the node to which each record is assigned.

The new field names are derived from the model name by adding prefixes. For C&R Tree, CHAID, and QUEST, the prefixes are \$R- for the prediction field, \$RC- for the confidence field, and \$RI- for the node identifier field. For C5.0 trees, the prefixes are \$C- for the prediction field and \$CC- for the confidence field. If multiple tree model nodes are present, the new field names will include numbers in the prefix to distinguish them if necessary—for example, \$R1- and \$RC1-, and \$R2-.

Working with Tree Model Nuggets

You can save or export information related to the model in a number of ways.

Note: Many of these options are also available from the tree builder window. From either the tree builder or a tree model nugget, you can:

- Generate a Filter or Select node based on the current tree. See [Generating Filter and Select Nodes](#) for more information.
- Generate a Rule Set nugget that represents the tree structure as a set of rules defining the terminal branches of the tree. See [Generating a Rule Set from a Decision Tree](#) for more information.
- In addition, for tree model nuggets only, you can export the model in PMML format. See [The models palette](#) for more information. If the model includes any custom splits, this information is not preserved in the exported PMML. (The split is preserved, but the fact that it is custom rather than chosen by the algorithm is not.)
- Generate a graph based on a selected part of the current tree. Note that this only works for a nugget when it is attached to other nodes in a stream. See [Generating Graphs](#) for more information.

- For boosted C5.0 models only, you can choose Single Decision Tree (Canvas) or Single Decision Tree (GM Palette) to create a new single rule set derived from the currently selected rule. See the topic [Boosted C5.0 Models](#) for more information.

Note: Although the Build Rule node was replaced by the C&R Tree node, decision tree nodes in existing streams that were originally created using a Build Rule node will still function properly.

- [Single Tree Model Nuggets](#)
- [Model nuggets for boosting, bagging, and very large datasets](#)

Related information

- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)
- [Decision Tree Model Rules](#)
- [Decision Tree Model Viewer](#)
- [Decision Tree/Rule Set model nugget settings](#)
- [Boosted C5.0 Models](#)
- [Generating Graphs](#)
- [C5.0 Node](#)
- [C&R Tree Node](#)
- [CHAID Node](#)
- [QUEST Node](#)

Single Tree Model Nuggets

If you select Build a single tree as the main objective on the modeling node, the resulting model nugget contains the following tabs.

Table 1. Tabs on single tree nugget

Tab	Description	Further Information
Model	Displays the rules that define the model.	See the topic Decision Tree Model Rules for more information.
Viewer	Displays the tree view of the model.	See the topic Decision Tree Model Viewer for more information.
Summary	Displays information about the fields, build settings, and model estimation process.	See the topic Model Nugget Summary/Information for more information.
Settings	Enables you to specify options for confidences and for SQL generation during model scoring.	See the topic Decision Tree/Rule Set model nugget settings for more information.
Annotation	Enables you to add descriptive annotations, specify a custom name, add tooltip text and specify search keywords for the model.	

- [Decision Tree Model Rules](#)
- [Decision Tree Model Viewer](#)
- [Decision Tree/Rule Set model nugget settings](#)
- [Boosted C5.0 Models](#)
- [Generating Graphs](#)

Decision Tree Model Rules

The Model tab for a decision tree nugget displays the rules that define the model. Optionally, a graph of predictor importance and a third panel with information about history, frequencies, and surrogates can also be displayed.

Note: If you select the Create a model for very large datasets option on the CHAID node Build Options tab (Objective panel), the Model tab displays the tree rule details only.

Tree Rules

The left pane displays a list of conditions defining the partitioning of data discovered by the algorithm—essentially a series of rules that can be used to assign individual records to child nodes based on the values of different predictors.

Decision trees work by recursively partitioning the data based on input field values. The data partitions are called *branches*. The initial branch (sometimes called the *root*) encompasses all data records. The root is split into subsets, or *child branches*, based on the value of a particular

input field. Each child branch can be further split into sub-branches, which can in turn be split again, and so on. At the lowest level of the tree are branches that have no more splits. Such branches are known as *terminal branches* (or *leaves*).

Tree rule details

The rule browser shows the input values that define each partition or branch and a summary of output field values for the records in that split. For general information on using the model browser, see [Browsing model nuggets](#).

For splits based on numeric fields, the branch is shown by a line of the form:

```
fieldname relation value [summary]
```

where *relation* is a numeric relation. For example, a branch defined by values greater than 100 for the *revenue* field would be shown as:

```
revenue > 100 [summary]
```

For splits based on symbolic fields, the branch is shown by a line of the form:

```
fieldname = value [summary] or fieldname in [values] [summary]
```

where *values* represents the field values that define the branch. For example, a branch that includes records where the value of *region* can be *North*, *West*, or *South* would be represented as:

```
region in ["North" "West" "South"] [summary]
```

For terminal branches, a prediction is also given, adding an arrow and the predicted value to the end of the rule condition. For example, a leaf defined by *revenue* > 100 that predicts a value of *high* for the output field would be displayed as:

```
revenue > 100 [Mode: high] → high
```

The *summary* for the branch is defined differently for symbolic and numeric output fields. For trees with numeric output fields, the summary is the *average* value for the branch, and the *effect* of the branch is the difference between the average for the branch and the average of its parent branch. For trees with symbolic output fields, the summary is the *mode*, or the most frequent value, for records in the branch.

To fully describe a branch, you need to include the condition that defines the branch, plus the conditions that define the splits further up the tree. For example, in the tree:

```
revenue > 100
region = "North"
region in ["South" "East" "West"]
revenue <= 200
```

the branch represented by the second line is defined by the conditions *revenue* > 100 and *region* = "North".

If you click Show Instances/Confidence on the toolbar, each rule will also show information about the number of records to which the rule applies (*Instances*) and the proportion of those records for which the rule is true (*Confidence*).

Predictor Importance

Optionally, a chart that indicates the relative importance of each predictor in estimating the model may also be displayed on the Model tab. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least.

Note: This chart is only available if Calculate predictor importance is selected on the Analyze tab before generating the model. See the topic [Predictor Importance](#) for more information.

Additional Model Information

If you click Show Additional Information Panel on the toolbar, you will see a panel at the bottom of the window displaying detailed information for the selected rule. The information panel contains three tabs.

History. This tab traces the split conditions from the root node down to the selected node. This provides a list of conditions that determine when a record is assigned to the selected node. Records for which all of the conditions are true will be assigned to this node.

Frequencies. For models with symbolic target fields, this tab shows, for each possible target value, the number of records assigned to this node (in the training data) that have that target value. The frequency figure, expressed as a percentage (shown to a maximum of three decimal places) is also displayed. For models with numeric targets, this tab is empty.

Surrogates. Where applicable, any surrogates for the primary split field are shown for the selected node. Surrogates are alternate fields used if the primary predictor value is missing for a given record. The maximum number of surrogates allowed for a given split is specified in the tree-building node, but the actual number depends on the training data. In general, the more missing data, the more surrogates are likely to be used. For other decision tree models, this tab is empty.

Note: To be included in the model, surrogates must be identified during the training phase. If the training sample has no missing values, then no surrogates will be identified, and any records with missing values encountered during testing or scoring will automatically fall into the child node with the largest number of records. If missing values are expected during testing or scoring, be sure that values are missing from the training sample, as well. Surrogates are not available for CHAID trees.

Effect

The effect of a node is the increase or decrease of the average value (predicted value compared to the parent node). For example, if the average of a node is 0.2 and the average of its parent is 0.6, then the effect for the node is $0.2 - 0.6 = -0.4$. This statistic only applies for a continuous target.

Decision Tree Model Viewer

The Viewer tab for a decision tree model nugget resembles the display in the tree builder. The main difference is that when browsing the model nugget, you can not grow or modify the tree. Other options for viewing and customizing the display are similar between the two components. See the topic [Customizing the Tree View](#) for more information.

Note: The Viewer tab is not displayed for CHAID model nuggets built if you select the Create a model for very large datasets option on the Build Options tab - Objective panel.

When viewing split rules on the Viewer tab, square brackets mean that the adjacent value is included in the range whereas parentheses indicate that the adjacent value is excluded from the range. The expression $(23, 37]$ therefore means from 23 exclusive to 37 inclusive, i.e. from just above 23 to 37. On the Model tab, the same condition would be displayed as:

`Age > 23 and Age <= 37`

Decision Tree/Rule Set model nugget settings

The Settings tab for a decision tree or Rule Set model nugget enables you to specify options for confidences and for SQL generation during model scoring. This tab is available only after the model nugget has been added to a stream.

Calculate confidences Select to include confidences in scoring operations. When scoring models in the database, excluding confidences enables you to generate more efficient SQL. For regression trees, confidences are not assigned.

Note: If you select the Create a model for very large datasets option on the Build Options tab - Method panel for CHAID models, this checkbox is available only in the model nuggets for categorical targets of nominal or flag.

Calculate raw propensity scores For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

Note: If you select the Create a model for very large datasets option on the Build Options tab - Method panel for CHAID models, this checkbox is available only in model nuggets with a categorical target of flag.

Calculate adjusted propensity scores Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

Note: Adjusted propensity scores are not available for boosted tree and rule set models. See the topic [Boosted C5.0 Models](#) for more information.

Rule identifier For CHAID, QUEST, and C&R Tree models, this option adds a field in the scoring output that indicates the ID for the terminal node to which each record is assigned.

Note: When this option is selected, SQL generation is not available.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default:** Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- **Score by converting to native SQL without missing value support** If selected, generates native SQL to score the model within the database, without the overhead of handling missing values. This option simply sets the prediction to null (`$null$`) when a missing value is encountered while scoring a case.

Note: This option is not available for CHAID models. For other model types, it is only available for decision trees (not rule sets).

- **Score by converting to native SQL with missing value support** For CHAID, QUEST, and C&R Tree models, you can generate native SQL to score the model within the database with full missing value support. This means that SQL is generated so that missing values are handled as specified in the model. For example, C&R Trees use surrogate rules and biggest child fallback.

Note: For C5.0 models, this option is only available for rule sets (not decision trees).

- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Boosted C5.0 Models

This feature is available in SPSS® Modeler Professional and SPSS Modeler Premium.

When you create a boosted C5.0 model (either a rule set or a decision tree), you actually create a set of related models. The model rule browser for a boosted C5.0 model shows the list of models at the top level of the hierarchy, along with the estimated accuracy of each model and the overall accuracy of the ensemble of boosted models. To examine the rules or splits for a particular model, select that model and expand it as you would a rule or branch in a single model.

You can also extract a particular model from the set of boosted models and create a new Rule Set model nugget containing just that model. To create a new rule set from a boosted C5.0 model, select the rule set or tree of interest and choose either Single Decision Tree (GM Palette) or Single Decision Tree (Canvas) from the Generate menu.

Related information

- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)
- [C&R Tree, CHAID, QUEST, and C5.0 decision tree model nuggets](#)
- [Decision Tree Model Rules](#)
- [Decision Tree Model Viewer](#)
- [Decision Tree/Rule Set model nugget settings](#)
- [Generating Graphs](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)
- [C5.0 Node](#)

Generating Graphs

The Tree nodes provide a lot of information; however, it may not always be in an easily accessible format for business users. To provide the data in a way that can be easily incorporated into business reports, presentations, and so on, you can produce graphs of selected data. For example, from either the Model or the Viewer tabs of a model nugget, or from the Viewer tab of an interactive tree, you can generate a graph for a selected part of the tree, thereby only creating a graph for the cases in the selected tree or branch node.

Note: You can only generate a graph from a nugget when it is attached to other nodes in a stream.

Generate a graph

The first step is to select the information to be shown on the graph:

- On the Model tab of a nugget, expand the list of conditions and rules in the left pane and select the one in which you are interested.
- On the Viewer tab of a nugget, expand the list of branches and select the node in which you are interested.
- On the Viewer tab of an interactive tree, expand the list of branches and select the node in which you are interested.

Note: You cannot select the top node on either Viewer tab.

The way in which you create a graph is the same, regardless of how you select the data to be shown:

1. From the Generate menu, select Graph (from selection); alternatively, on the Viewer tab, click the Graph (from selection) button in the bottom left corner. The Graphboard Basic tab is displayed.
Note: Only the Basic and Detailed tabs are available when you display the Graphboard in this way.
2. Using either the Basic or Detailed tab settings, specify the details to be displayed on the graph.
3. Click OK to generate the graph.

The graph heading identifies the nodes or rules that were chosen for inclusion.

Related information

- [Generating Filter and Select Nodes](#)
- [Generating a Rule Set from a Decision Tree](#)
- [C&R Tree, CHAID, QUEST, and C5.0 decision tree model nuggets](#)
- [Decision Tree Model Rules](#)
- [Decision Tree Model Viewer](#)
- [Decision Tree/Rule Set model nugget settings](#)

- [Boosted C5.0 Models](#)
- [C5.0 Node](#)
- [C&R Tree Node](#)
- [CHAID Node](#)
- [QUEST Node](#)
- [The Interactive Tree Builder](#)

Model nuggets for boosting, bagging, and very large datasets

If you select Enhance model accuracy (boosting), Enhance model stability (bagging), or Create a model for very large datasets as the main objective on the modeling node, IBM® SPSS® Modeler builds an ensemble of multiple models. See the topic [Models for ensembles](#) for more information.

The resulting model nugget contains the following tabs. The Model tab provides a number of different views of the model.

Table 1. Tabs available in model nugget

Tab	View	Description	Further Information
Model	Model Summary	Displays a summary of the ensemble quality and (except for boosted models and continuous targets) diversity, a measure of how much the predictions vary across the different models.	See the topic Model Summary (ensemble viewer) for more information.
	Predictor Importance	Displays a chart indicating the relative importance of each predictor (input field) in estimating the model.	See the topic Predictor Importance (ensemble viewer) for more information.
	Predictor Frequency	Displays a chart showing the relative frequency with which each predictor is used in the set of models.	See the topic Predictor Frequency (ensemble viewer) for more information.
	Component Model Accuracy	Plots a chart of the predictive accuracy of each of the different models in the ensemble.	
	Component Model Details	Displays information on each of the different models in the ensemble.	See the topic Component Model Details (ensemble viewer) for more information.
	Information	Displays information about the fields, build settings, and model estimation process.	See the topic Model Nugget Summary / Information for more information.
Settings		Enables you to include confidences in scoring operations.	See the topic Decision Tree/Rule Set model nugget settings for more information.
Annotation		Enables you to add descriptive annotations, specify a custom name, add tooltip text and specify search keywords for the model.	

C&R Tree, CHAID, QUEST, C5.0, and Apriori rule set model nuggets

A Rule Set model nugget represents the rules for predicting a particular output field discovered by the association rule modeling node (Apriori) or by one of the tree-building nodes (C&R Tree, CHAID, QUEST, or C5.0). For association rules, the rule set must be generated from an unrefined Rule nugget. For trees, a rule set can be generated from the interactive tree builder, from a C5.0 model-building node, or from any tree model nugget. Unlike unrefined Rule nuggets, Rule Set nuggets can be placed in streams to generate predictions.

When you run a stream containing a Rule Set nugget, two new fields are added to the stream containing the predicted value and the confidence for each record to the data. The new field names are derived from the model name by adding prefixes. For association rule sets, the prefixes are \$A- for the prediction field and \$AC- for the confidence field. For C5.0 rule sets, the prefixes are \$C- for the prediction field and \$CC- for the confidence field. For C&R Tree rule sets, the prefixes are \$R- for the prediction field and \$RC- for the confidence field. In a stream with multiple Rule Set nuggets in a series predicting the same output field(s), the new field names will include numbers in the prefix to distinguish them from each other. The first association Rule Set nugget in the stream will use the usual names, the second node will use names starting with \$A1- and \$AC1-, the third node will use names starting with \$A2- and \$AC2-, and so on.

How rules are applied. Rule Sets generated from association rules are unlike other model nuggets because for any particular record, more than one prediction can be generated, and those predictions may not all agree. There are two methods for generating predictions from rule sets.

Note: Rule Sets that are generated from decision trees return the same results regardless of which method is used, since the rules derived from a decision tree are mutually exclusive.

- **Voting.** This method attempts to combine the predictions of all of the rules that apply to the record. For each record, all rules are examined and each rule that applies to the record is used to generate a prediction and an associated confidence. The sum of confidence

figures for each output value is computed, and the value with the greatest confidence sum is chosen as the final prediction. The confidence for the final prediction is the confidence sum for that value divided by the number of rules that fired for that record.

- **First hit.** This method simply tests the rules in order, and the first rule that applies to the record is the one used to generate the prediction.

The method used can be controlled in the stream options.

Generating nodes. The Generate menu enables you to create new nodes based on the rule set.

- Filter Node Creates a new Filter node to filter fields that are not used by rules in the rule set.
- Select Node Creates a new Select node to select records to which the selected rule applies. The generated node will select records for which all antecedents of the rule are true. This option requires a rule to be selected.
- Rule Trace Node Creates a new SuperNode that will compute a field indicating which rule was used to make the prediction for each record. When a rule set is evaluated using the first hit method, this is simply a symbol indicating the first rule that would fire. When the rule set is evaluated using the voting method, this is a more complex string showing the input to the voting mechanism.
- Single Decision Tree (Canvas) / Single Decision Tree (GM Palette). Creates a new single Rule Set nugget derived from the currently selected rule. Available only for **boosted** C5.0 models. See the topic [Boosted C5.0 Models](#) for more information.
- Model to Palette Returns the model to the models palette. This is useful in situations where a colleague may have sent you a stream containing the model and not the model itself.

Note: The Settings and Summary tabs in the Rule Set nugget are identical to those for decision tree models.

- [Rule Set Model Tab](#)

Related information

- [Decision Tree/Rule Set model nugget settings](#)
- [Rule Set Model Tab](#)
- [Association Rule Model Nugget Settings](#)
- [C5.0 Node](#)
- [C&R Tree Node](#)
- [CHAID Node](#)
- [QUEST Node](#)
- [Boosted C5.0 Models](#)

Rule Set Model Tab

The Model tab for a Rule Set nugget displays a list of rules extracted from the data by the algorithm.

Rules are broken down by consequent (predicted category) and are presented in the following format:

```
if antecedent_1  
and antecedent_2  
...  
and antecedent_n  
then predicted value
```

where **consequent** and **antecedent_1** through **antecedent_n** are all conditions. The rule is interpreted as "for records where **antecedent_1** through **antecedent_n** are all true, **consequent** is also likely to be true." If you click the Show Instances/Confidence button on the toolbar, each rule will also show information on the number of records to which the rule applies--that is, for which the antecedents are true (**Instances**) and the proportion of those records for which the entire rule is true (**Confidence**).

Note that confidence is calculated somewhat differently for C5.0 rule sets. C5.0 uses the following formula for calculating the confidence of a rule:

```
(1 + number of records where rule is correct)  
/  
(2 + number of records for which the rule's antecedents are true)
```

This calculation of the confidence estimate adjusts for the process of generalizing rules from a decision tree (which is what C5.0 does when it creates a rule set).

Related information

- [Decision Tree/Rule Set model nugget settings](#)
- [C&R Tree, CHAID, QUEST, C5.0, and Apriori rule set model nuggets](#)
- [Association Rule Model Nugget Settings](#)
- [C5.0 Node](#)
- [C&R Tree Node](#)

- [CHAID Node](#)
- [QUEST Node](#)
- [Boosted C5.0 Models](#)

Bayesian Network Models

- [Bayesian Network Node](#)
- [Bayesian Network Model Nuggets](#)

Bayesian Network Node

The **Bayesian Network** node enables you to build a probability model by combining observed and recorded evidence with "common-sense" real-world knowledge to establish the likelihood of occurrences by using seemingly unlinked attributes. The node focuses on Tree Augmented Naïve Bayes (TAN) and Markov Blanket networks that are primarily used for classification.

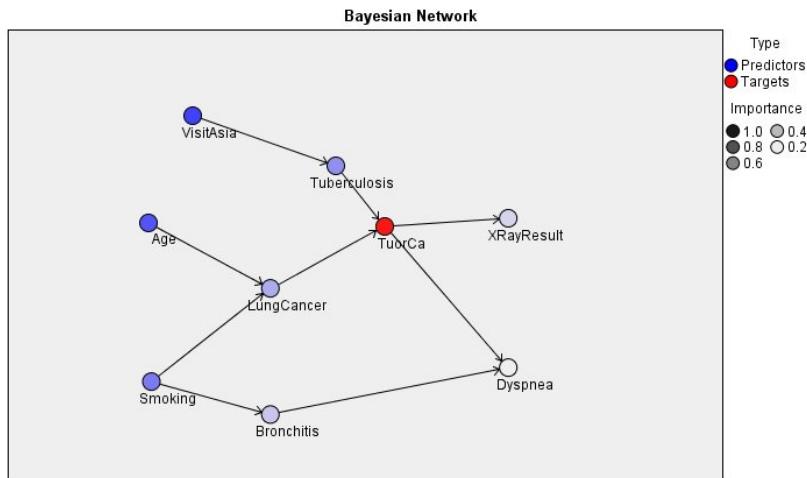
Bayesian networks are used for making predictions in many varied situations; some examples are:

- Selecting loan opportunities with low default risk.
- Estimating when equipment will need service, parts, or replacement, based on sensor input and existing records.
- Resolving customer problems via online troubleshooting tools.
- Diagnosing and troubleshooting cellular telephone networks in real-time.
- Assessing the potential risks and rewards of research-and-development projects in order to focus resources on the best opportunities.

A Bayesian network is a graphical model that displays variables (often referred to as **nodes**) in a dataset and the probabilistic, or conditional, independencies between them. Causal relationships between nodes may be represented by a Bayesian network; however, the links in the network (also known as **arcs**) do not necessarily represent direct cause and effect. For example, a Bayesian network can be used to calculate the probability of a patient having a specific disease, given the presence or absence of certain symptoms and other relevant data, if the probabilistic independencies between symptoms and disease as displayed on the graph hold true. Networks are very robust where information is missing and make the best possible prediction using whatever information is present.

A common, basic, example of a Bayesian network was created by Lauritzen and Spiegelhalter (1988). It is often referred to as the "Asia" model and is a simplified version of a network that may be used to diagnose a doctor's new patients; the direction of the links roughly corresponding to causality. Each node represents a facet that may relate to the patient's condition; for example, "Smoking" indicates that they are a confirmed smoker, and "VisitAsia" shows if they recently visited Asia. Probability relationships are shown by the links between any nodes; for example, smoking increases the chances of the patient developing both bronchitis and lung cancer, whereas age only seems to be associated with the possibility of developing lung cancer. In the same way, abnormalities on an x-ray of the lungs may be caused by either tuberculosis or lung cancer, while the chances of a patient suffering from shortness of breath (dyspnea) are increased if they also suffer from either bronchitis or lung cancer.

Figure 1. Lauritzen and Spiegelhalter's Asia network example



There are several reasons why you might decide to use a Bayesian network:

- It helps you learn about causal relationships. From this, it enables you to understand a problem area and to predict the consequences of any intervention.
- The network provides an efficient approach for avoiding the overfitting of data.
- A clear visualization of the relationships involved is easily observed.

Requirements. Target fields must be categorical and can have a measurement level of *Nominal*, *Ordinal*, or *Flag*. Inputs can be fields of any type. Continuous (numeric range) input fields will be automatically binned; however, if the distribution is skewed, you may obtain better results by manually binning the fields using a Binning node before the Bayesian Network node. For example, use Optimal Binning where the Supervisor field is the same as the Bayesian Network node Target field.

Example. An analyst for a bank wants to be able to predict customers, or potential customers, who are likely to default on their loan repayments. You can use a Bayesian network model to identify the characteristics of customers most likely to default, and build several different types of model to establish which is the best at predicting potential defaulters.

Example. A telecommunications operator wants to reduce the number of customers who leave the business (known as "churn"), and update the model on a monthly basis using each preceding month's data. You can use a Bayesian network model to identify the characteristics of customers most likely to churn, and continue training the model each month with the new data.

- [Bayesian Network Node Model Options](#)
- [Bayesian Network Node Expert Options](#)

Related information

- [Bayesian Network Node Model Options](#)
 - [Bayesian Network Node Expert Options](#)
 - [Bayesian Network Model Nuggets](#)
 - [Bayesian Network Model Settings](#)
 - [Bayesian Network Model Summary](#)
-

Bayesian Network Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Build model for each split. Builds a separate model for each possible value of input fields that are specified as split fields. See the topic [Building Split Models](#) for more information.

Partition. This field enables you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to generate the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)

Splits. For split models, select the split field or fields. This is similar to setting the field role to *Split* in a Type node. You can designate only fields with a measurement level of Flag, Nominal, Ordinal or Continuous as split fields. Fields chosen as split fields cannot be used as target, input, partition, frequency or weight fields. See the topic [Building Split Models](#) for more information.

Continue training existing model. If you select this option, the results shown on the model nugget Model tab are regenerated and updated each time the model is run. For example, you would do this when you have added a new or updated data source to an existing model.

Note: This can only update the existing network; it cannot add or remove nodes or connections. Each time you retrain the model the network will be the same shape, only the conditional probabilities and predictor importance will change. If your new data are broadly similar to your old data then this does not matter since you expect the same things to be significant; however, if you want to check or update *what* is significant (as opposed to how significant it is), you will need to build a new model, that is, build a new network

Structure type. Select the structure to be used when building the Bayesian network:

- **TAN.** The Tree Augmented Naïve Bayes model (TAN) creates a simple Bayesian network model that is an improvement over the standard Naïve Bayes model. This is because it allows each predictor to depend on another predictor in addition to the target variable, thereby increasing the classification accuracy.
- **Markov Blanket.** This selects the set of nodes in the dataset that contain the target variable's parents, its children, and its children's parents. Essentially, a Markov blanket identifies all the variables in the network that are needed to predict the target variable. This method of building a network is considered to be more accurate; however, with large datasets there maybe a processing time-penalty due to the high number of variables involved. To reduce the amount of processing, you can use the Feature Selection options on the Expert tab to select the variables that are significantly related to the target variable.

Include feature selection preprocessing step. Selecting this box enables you to use the Feature Selection options on the Expert tab.

Parameter learning method. Bayesian network parameters refer to the conditional probabilities for each node given the values of its parents. There are two possible selections that you can use to control the task of estimating the conditional probability tables between nodes where the values of the parents are known:

- **Maximum likelihood.** Select this box when using a large dataset. This is the default selection.
- **Bayes adjustment for small cell counts.** For smaller datasets there is a danger of overfitting the model, as well as the possibility of a high number of zero-counts. Select this option to alleviate these problems by applying smoothing to reduce the effect of any zero-counts and any unreliable estimate effects.

Related information

- [Bayesian Network Node](#)
 - [Bayesian Network Node Expert Options](#)
 - [Bayesian Network Model Nuggets](#)
 - [Bayesian Network Model Settings](#)
 - [Bayesian Network Model Summary](#)
-

Bayesian Network Node Expert Options

The node expert options enable you to fine-tune the model-building process. To access the expert options, set Mode to Expert on the Expert tab.

Missing values. By default, IBM® SPSS® Modeler only uses records that have valid values for all fields used in the model. (This is sometimes called **listwise deletion** of missing values.) If you have a lot of missing data, you may find that this approach eliminates too many records, leaving you without enough data to generate a good model. In such cases, you can deselect the Use only complete records option. IBM SPSS Modeler then attempts to use as much information as possible to estimate the model, including records where some of the fields have missing values. (This is sometimes called **pairwise deletion** of missing values.) However, in some situations, using incomplete records in this manner can lead to computational problems in estimating the model.

Append all probabilities. Specifies whether probabilities for each category of the output field are added to each record processed by the node. If this option is not selected, the probability of only the predicted category is added.

Independence test. A test of independence assesses whether paired observations on two variables are independent of each other. Select the type of test to be used, available options are:

- Likelihood ratio. Tests for target-predictor independence by calculating a ratio between the maximum probability of a result under two different hypotheses.
- Pearson chi-square. Tests for target-predictor independence by using a null hypothesis that the relative frequencies of occurrence of observed events follow a specified frequency distribution.

Bayesian network models conduct conditional tests of independence where additional variables are used beyond the tested pairs. In addition, the models explore not only the relations between the target and predictors, but also the relations among the predictors themselves.

Note: The Independence test options are only available if you select either Include feature selection preprocessing step or a Structure type of Markov Blanket on the Model tab.

Significance level. Used in conjunction with the Independence test settings, this enables you to set a cut-off value to be used when conducting the tests. The lower the value, the fewer links remain in the network; the default level is 0.01.

Note: This option is only available if you select either Include feature selection preprocessing step or a Structure type of Markov Blanket on the Model tab.

Maximal conditioning set size. The algorithm for creating a Markov Blanket structure uses conditioning sets of increasing size to carry out independence testing and remove unnecessary links from the network. Because tests involving a high number of conditioning variables require more time and memory for processing you can limit the number of variables to be included. This can be especially useful when processing data with strong dependencies among many variables. Note however that the resulting network may contain some superfluous links.

Specify the maximal number of conditioning variables to be used for independence testing. The default setting is 5.

Note: This option is only available if you select either Include feature selection preprocessing step or a Structure type of Markov Blanket on the Model tab.

Feature selection. These options enable you to restrict the number of inputs used when processing the model in order to speed up the model building process. This is especially useful when creating a Markov Blanket structure due to the possible large number of potential inputs; it enables you to select the inputs that are significantly related to the target variable.

Note: The feature selection options are only available if you select Include feature selection preprocessing step on the Model tab.

- **Inputs always selected.** Using the Field Chooser (button to the right of the text field), select the fields from the dataset that are always to be used when building the Bayesian network model. The target field is always selected. Note that the Bayesian network may still drop the items from this list during the model building process if other tests do not consider it significant. So this option simply ensures that the items in the list are used in the model building process itself, not that they will absolutely appear in the resulting Bayesian model.

- Maximum number of inputs. Specify the total number of inputs from the dataset to be used when building the Bayesian network model. The highest number you can enter is the total number of inputs in the dataset.
Note: If the number of fields selected in Inputs always selected exceeds the value of Maximum number of inputs, an error message is displayed.

Related information

- [Bayesian Network Node](#)
 - [Bayesian Network Node Model Options](#)
 - [Bayesian Network Model Nuggets](#)
 - [Bayesian Network Model Settings](#)
 - [Bayesian Network Model Summary](#)
-

Bayesian Network Model Nuggets

Note: If you select Continue training existing parameters on the modeling node Model tab, the information that is shown on the model nugget Model tab is updated each time that you regenerate the model.

The model nugget Model tab is split into two panes.

Left Pane

This view contains a network graph of nodes that displays the relationship between the target and its most important predictors, and the relationship between the predictors. The importance of each predictor is shown by the density of its color; a strong color shows an important predictor, and vice versa.

The bin values for nodes that represent a range are displayed in a tooltip when you hover the mouse pointer over the node.

You can use the graph tools in IBM® SPSS® Modeler to interact, edit, and save the graph. For example, for use in other applications such as MS Word.

Tip: If the network contains many nodes, you can click to select a node and drag it to make the graph more legible.

Distribution This view displays the conditional probabilities for each node in the network as a mini graph. Hover the mouse pointer over a graph to display its values in a tooltip.

Right Pane

Predictor Importance This displays a chart that indicates the relative importance of each predictor in estimating the model. For more information, see [Predictor Importance](#).

Conditional Probabilities When you select a node or mini distribution graph in the left pane the associated conditional probabilities table is displayed in the right pane. This table contains the conditional probability value for each node value and each combination of values in its parent nodes. In addition, it includes the number of records that are observed for each record value and each combination of values in the parent nodes.

- [Bayesian Network Model Settings](#)
- [Bayesian Network Model Summary](#)

Related information

- [Bayesian Network Node](#)
 - [Bayesian Network Node Model Options](#)
 - [Bayesian Network Node Expert Options](#)
 - [Bayesian Network Model Settings](#)
 - [Bayesian Network Model Summary](#)
-

Bayesian Network Model Settings

The Settings tab for a Bayesian Network model nugget specifies options for modifying the built model. For example, you may use the Bayesian Network node to build several different models using the same data and settings, then use this tab in each model to slightly modify the settings to see how that affects the results.

Note: This tab is only available after the model nugget has been added to a stream.

Calculate raw propensity scores. For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

Calculate adjusted propensity scores. Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

Append all probabilities Specifies whether probabilities for each category of the output field are added to each record processed by the node. If this option is not selected, the probability of only the predicted category is added.

The default setting of this check box is determined by the corresponding check box on the Expert tab of the modeling node. See the topic [Bayesian Network Node Expert Options](#) for more information.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Related information

- [Bayesian Network Node](#)
 - [Bayesian Network Node Model Options](#)
 - [Bayesian Network Node Expert Options](#)
 - [Bayesian Network Model Nuggets](#)
 - [Bayesian Network Model Summary](#)
-

Bayesian Network Model Summary

The Summary tab of a model nugget displays information about the model itself (*Analysis*), fields used in the model (*Fields*), settings used when building the model (*Build Settings*), and model training (*Training Summary*).

When you first browse the node, the Summary tab results are collapsed. To see the results of interest, use the expander control to the left of an item to unfold it or click the Expand All button to show all results. To hide the results when you have finished viewing them, use the expander control to collapse the specific results that you want to hide or click the Collapse All button to collapse all results.

Analysis. Displays information about the specific model.

Fields. Lists the fields used as the target and the inputs in building the model.

Build Settings. Contains information about the settings used in building the model.

Training Summary. Shows the type of model, the stream used to create it, the user who created it, when it was built, and the elapsed time for building the model.

Related information

- [Bayesian Network Node](#)
 - [Bayesian Network Node Model Options](#)
 - [Bayesian Network Node Expert Options](#)
 - [Bayesian Network Model Nuggets](#)
 - [Bayesian Network Model Settings](#)
-

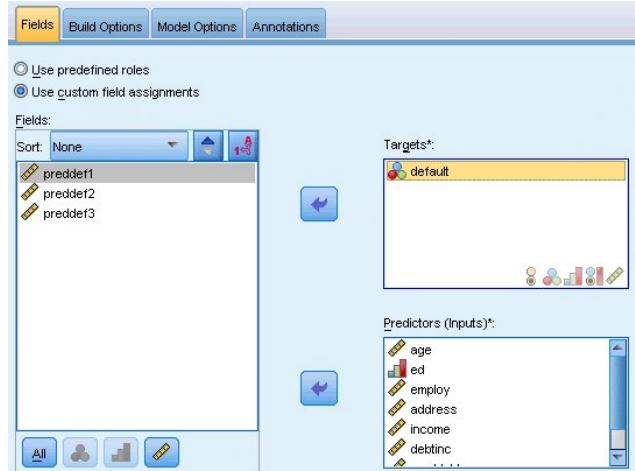
Neural networks

A **neural network** can approximate a wide range of predictive models with minimal demands on model structure and assumption. The form of the relationships is determined during the learning process. If a linear relationship between the target and predictors is appropriate, the results

of the neural network should closely approximate those of a traditional linear model. If a nonlinear relationship is more appropriate, the neural network will automatically approximate the "correct" model structure.

The trade-off for this flexibility is that the neural network is not easily interpretable. If you are trying to explain an underlying process that produces the relationships between the target and predictors, it would be better to use a more traditional statistical model. However, if model interpretability is not important, you can obtain good predictions using a neural network.

Figure 1. Fields tab



Field requirements. There must be at least one Target and one Input. Fields set to Both or None are ignored. There are no measurement level restrictions on targets or predictors (inputs). See [Modeling Node Fields Options](#) for more information.

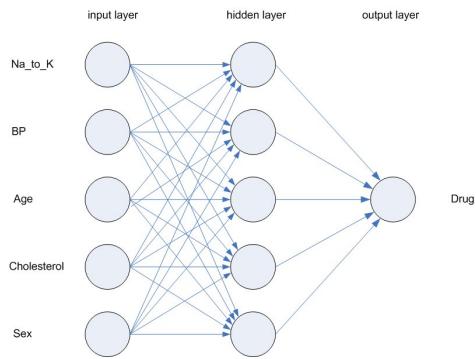
The initial weights assigned to neural networks during model building, and therefore the final models produced, depend on the order of the fields in the data. SPSS® Modeler automatically sorts data by field name before presenting it to the neural network for training. This means that explicitly changing the order of the fields in the data upstream will not affect the generated neural net models when a random seed is set in the model builder. However, changing the input field names in a way that changes their sort order will produce different neural network models, even with a random seed set in the model builder. The model quality will not be affected significantly given different sort order of field names.

- [The neural networks model](#)
- [Using neural networks with legacy streams](#)
- [Objectives \(neural networks\)](#)
- [Basics \(neural networks\)](#)
- [Stopping rules \(neural networks\)](#)
- [Ensembles \(neural networks\)](#)
- [Advanced \(neural networks\)](#)
- [Model Options \(neural networks\)](#)
- [Model Summary \(neural networks\)](#)
- [Predictor Importance \(neural networks\)](#)
- [Predicted By Observed \(neural networks\)](#)
- [Classification \(neural networks\)](#)
- [Network \(neural networks\)](#)
- [Settings \(neural networks\)](#)
- [Neural Net Node](#)
- [Neural Net Model Nuggets](#)

The neural networks model

Neural networks are simple models of the way the nervous system operates. The basic units are **neurons**, which are typically organized into **layers**, as shown in the following figure.

Figure 1. Structure of a neural network



A **neural network** is a simplified model of the way the human brain processes information. It works by simulating a large number of interconnected processing units that resemble abstract versions of neurons.

The processing units are arranged in layers. There are typically three parts in a neural network: an **input layer**, with units representing the input fields; one or more **hidden layers**; and an **output layer**, with a unit or units representing the target field(s). The units are connected with varying connection strengths (or **weights**). Input data are presented to the first layer, and values are propagated from each neuron to every neuron in the next layer. Eventually, a result is delivered from the output layer.

The network learns by examining individual records, generating a prediction for each record, and making adjustments to the weights whenever it makes an incorrect prediction. This process is repeated many times, and the network continues to improve its predictions until one or more of the stopping criteria have been met.

Initially, all weights are random, and the answers that come out of the net are probably nonsensical. The network learns through **training**. Examples for which the output is known are repeatedly presented to the network, and the answers it gives are compared to the known outcomes. Information from this comparison is passed back through the network, gradually changing the weights. As training progresses, the network becomes increasingly accurate in replicating the known outcomes. Once trained, the network can be applied to future cases where the outcome is unknown.

Using neural networks with legacy streams

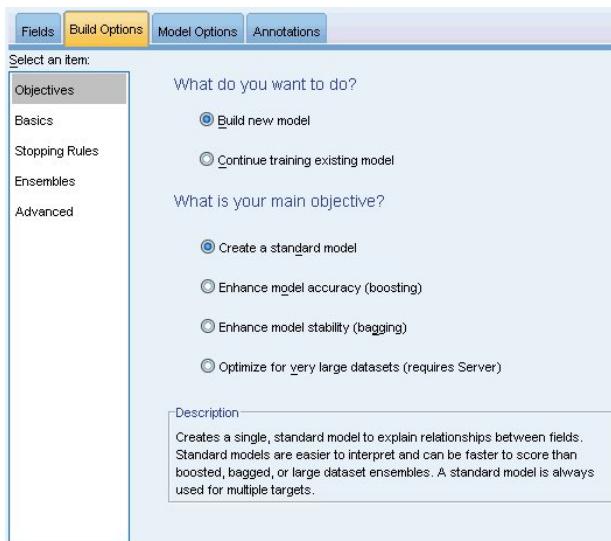
Version 14 of IBM® SPSS® Modeler introduced a new Neural Net node, supporting boosting and bagging techniques and optimization for very large datasets. Existing streams containing the old node may still build and score models in later releases. However, this support will be removed in a future release, so we recommend using the new version.

From version 13 onwards, fields with unknown values (that is, values not present in the training data) are no longer automatically treated as missing values, and are scored with the value `$null$`. Thus if you want to score fields with unknown values as non-null using an older (pre-13) Neural Net model in version 13 or later, you should mark unknown values as missing values (for example, by means of the Type node).

Note that, for compatibility, any legacy streams that still contain the old node may still be using the *Limit set size* option from Tools > Stream Properties > Options; this option only applies to Kohonen nets and K-Means nodes from version 14 onwards.

Objectives (neural networks)

Figure 1. Objectives settings



What do you want to do?

- Build a new model. Build a completely new model. This is the usual operation of the node.
 - Continue training an existing model. Training continues with the last model successfully produced by the node. This makes it possible to update or refresh an existing model without having to access the original data and may result in significantly faster performance since only the new or updated records are fed into the stream. Details on the previous model are stored with the modeling node, making it possible to use this option even if the previous model nugget is no longer available in the stream or Models palette.
- Note: When this option is enabled, all other controls on the Fields and Build Options tabs are disabled.

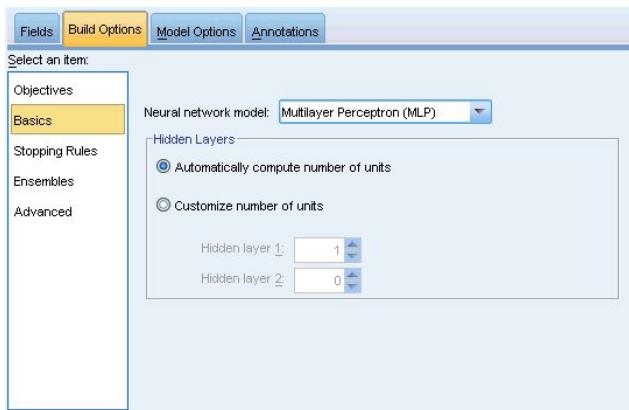
What is your main objective? Select the appropriate objective.

- Create a standard model. The method builds a single model to predict the target using the predictors. Generally speaking, standard models are easier to interpret and can be faster to score than boosted, bagged, or large dataset ensembles.
Note: Continue training existing model is only supported with Build a single tree split models, and you must be connected to Analytic Server.
- Enhance model accuracy (boosting). The method builds an ensemble model using boosting, which generates a sequence of models to obtain more accurate predictions. Ensembles can take longer to build and to score than a standard model.
Boosting produces a succession of "component models", each of which is built on the entire dataset. Prior to building each successive component model, the records are weighted based on the previous component model's residuals. Cases with large residuals are given relatively higher analysis weights so that the next component model will focus on predicting these records well. Together these component models form an ensemble model. The ensemble model scores new records using a combining rule; the available rules depend upon the measurement level of the target.
- Enhance model stability (bagging). The method builds an ensemble model using bagging (bootstrap aggregating), which generates multiple models to obtain more reliable predictions. Ensembles can take longer to build and to score than a standard model.
Bootstrap aggregation (bagging) produces replicates of the training dataset by sampling with replacement from the original dataset. This creates bootstrap samples of equal size to the original dataset. Then a "component model" is built on each replicate. Together these component models form an ensemble model. The ensemble model scores new records using a combining rule; the available rules depend upon the measurement level of the target.
- Create a model for very large datasets. The method builds an ensemble model by splitting the dataset into separate data blocks. Choose this option if your dataset is too large to build any of the models above, or for incremental model building. This option can take less time to build, but can take longer to score than a standard model.

When there are multiple targets, this method will only create a standard model, regardless of the selected objective.

Basics (neural networks)

Figure 1. Basics settings



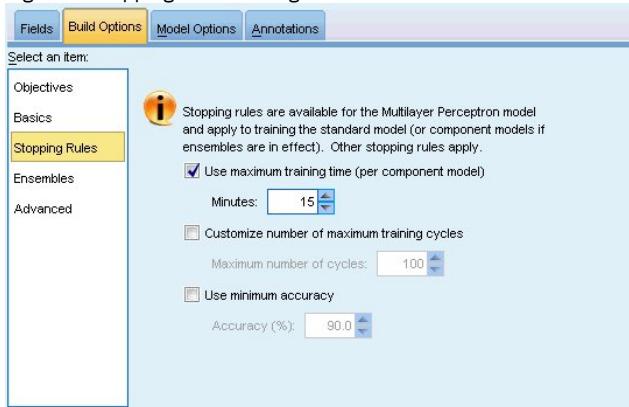
Neural network model. The type of model determines how the network connects the predictors to the targets through the hidden layer(s). The **multilayer perceptron (MLP)** allows for more complex relationships at the possible cost of increasing the training and scoring time. The **radial basis function (RBF)** may have lower training and scoring times, at the possible cost of reduced predictive power compared to the MLP.

Hidden Layers. The hidden layer(s) of a neural network contains unobservable units. The value of each hidden unit is some function of the predictors; the exact form of the function depends in part upon the network type. A multilayer perceptron can have one or two hidden layers; a radial basis function network can have one hidden layer.

- Automatically compute number of units. This option builds a network with one hidden layer and computes the "best" number of units in the hidden layer.
 - Customize number of units. This option allows you to specify the number of units in each hidden layer. The first hidden layer must have at least one unit. Specifying 0 units for the second hidden layer builds a multilayer perceptron with a single hidden layer.
- Note: You should choose values so that the number of nodes does not exceed the number of continuous predictors plus the total number of categories across all categorical (flag, nominal, and ordinal) predictors.

Stopping rules (neural networks)

Figure 1. Stopping Rules settings



These are the rules that determine when to stop training multilayer perceptron networks; these settings are ignored when the radial basis function algorithm is used. Training proceeds through at least one cycle (data pass), and can then be stopped according to the following criteria.

Use maximum training time (per component model). Choose whether to specify a maximum number of minutes for the algorithm to run. Specify a number greater than 0. When an ensemble model is built, this is the training time allowed for each component model of the ensemble. Note that training may go a bit beyond the specified time limit in order to complete the current cycle.

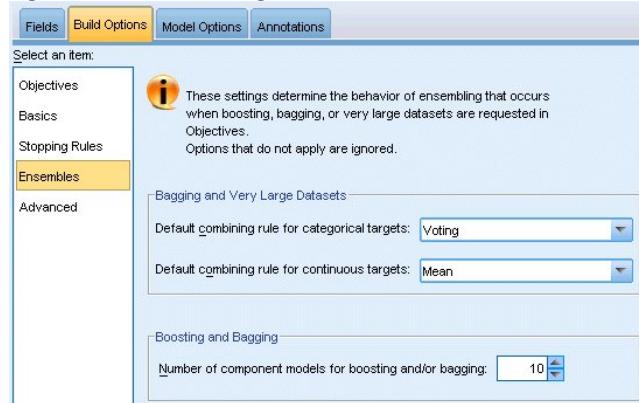
Customize number of maximum training cycles. The maximum number of training cycles allowed. If the maximum number of cycles is exceeded, then training stops. Specify an integer greater than 0.

Use minimum accuracy. With this option, training will continue until the specified accuracy is attained. This may never happen, but you can interrupt training at any point and save the net with the best accuracy achieved so far.

The training algorithm will also stop if the error in the overfit prevention set does not decrease after each cycle, if the relative change in the training error is small, or if the ratio of the current training error is small compared to the initial error.

Ensembles (neural networks)

Figure 1. Ensembles settings



These settings determine the behavior of ensembling that occurs when boosting, bagging, or very large datasets are requested in Objectives. Options that do not apply to the selected objective are ignored.

Bagging and Very Large Datasets. When scoring an ensemble, this is the rule used to combine the predicted values from the base models to compute the ensemble score value.

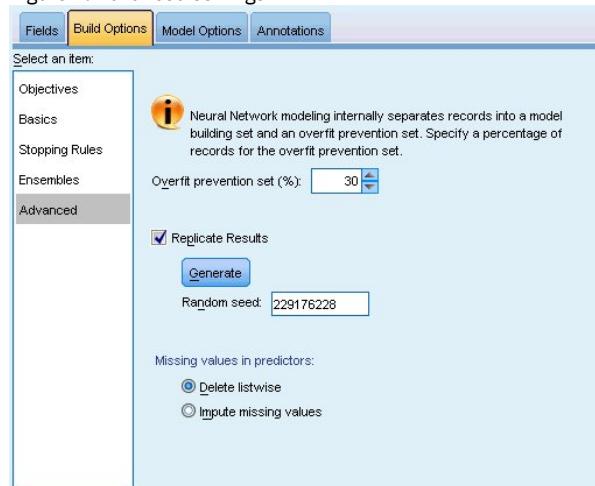
- Default combining rule for categorical targets. Ensemble predicted values for categorical targets can be combined using voting, highest probability, or highest mean probability. **Voting** selects the category that has the highest probability most often across the base models. **Highest probability** selects the category that achieves the single highest probability across all base models. **Highest mean probability** selects the category with the highest value when the category probabilities are averaged across base models.
- Default combining rule for continuous targets. Ensemble predicted values for continuous targets can be combined using the mean or median of the predicted values from the base models.

Note that when the objective is to enhance model accuracy, the combining rule selections are ignored. Boosting always uses a weighted majority vote to score categorical targets and a weighted median to score continuous targets.

Boosting and Bagging. Specify the number of base models to build when the objective is to enhance model accuracy or stability; for bagging, this is the number of bootstrap samples. It should be a positive integer.

Advanced (neural networks)

Figure 1. Advanced settings



Advanced settings provide control over options that do not fit neatly into other groups of settings.

Overfit prevention set. The neural network method internally separates records into a model building set and an overfit prevention set, which is an independent set of data records used to track errors during training in order to prevent the method from modeling chance variation in the data. Specify a percentage of records. The default is 30.

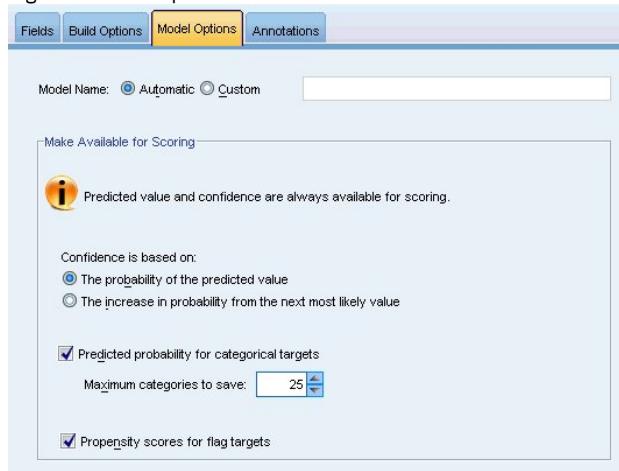
Replicate results. Setting a random seed allows you to replicate analyses. Specify an integer or click Generate, which will create a pseudo-random integer between 1 and 2147483647, inclusive. By default, analyses are replicated with seed 229176228.

Missing values in predictors. This specifies how to treat missing values. **Delete listwise** removes records with missing values on predictors from model building. **Impute missing values** will replace missing values in predictors and use those records in the analysis. Continuous fields impute

the average of the minimum and maximum observed values; categorical fields impute the most frequently occurring category. Note that records with missing values on any other field specified on the Fields tab are always removed from model building.

Model Options (neural networks)

Figure 1. Model Options tab



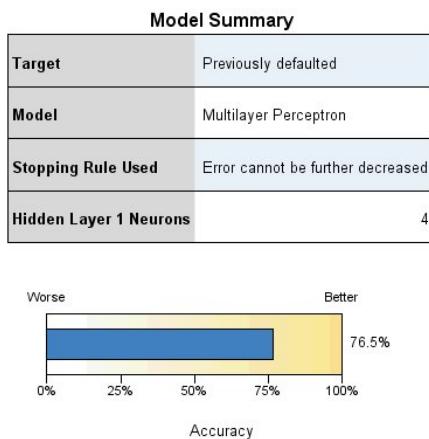
Model Name. You can generate the model name automatically based on the target fields or specify a custom name. The automatically generated name is the target field name. If there are multiple targets, then the model name is the field names in order, connected by ampersands. For example, if *field1* *field2* *field3* are targets, then the model name is: *field1 & field2 & field3*.

Make Available for Scoring. When the model is scored, the selected items in this group should be produced. The predicted value (for all targets) and confidence (for categorical targets) are always computed when the model is scored. The computed confidence can be based on the probability of the predicted value (the highest predicted probability) or the difference between the highest predicted probability and the second highest predicted probability.

- Predicted probability for categorical targets. This produces the predicted probabilities for categorical targets. A field is created for each category.
- Propensity scores for flag targets. For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. The model produces raw propensity scores; if partitions are in effect, the model also produces adjusted propensity scores based on the testing partition.

Model Summary (neural networks)

Figure 1. Neural Networks Model Summary view



The Model Summary view is a snapshot, at-a-glance summary of the neural network predictive or classification accuracy.

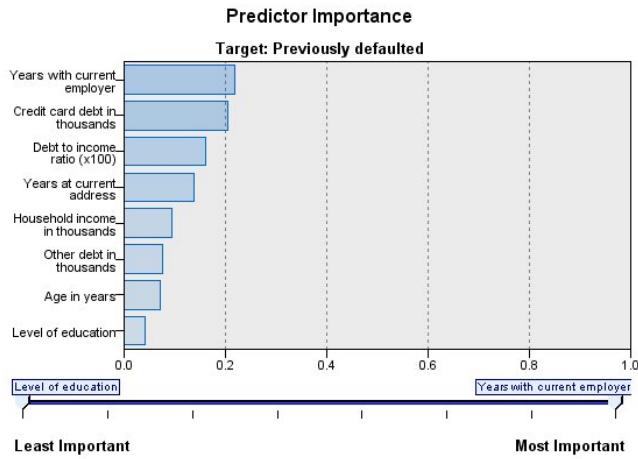
Model summary. The table identifies the target, the type of neural network trained, the stopping rule that stopped training (shown if a multilayer perceptron network was trained), and the number of neurons in each hidden layer of the network.

Neural Network Quality. The chart displays the accuracy of the final model, which is presented in larger is better format. For a categorical target, this is simply the percentage of records for which the predicted value matches the observed value. For a continuous target, the accuracy is given as the R² value.

Multiple targets. If there are multiple targets, then each target is displayed in the Target row of the table. The accuracy displayed in the chart is the average of the individual target accuracies.

Predictor Importance (neural networks)

Figure 1. Predictor Importance view

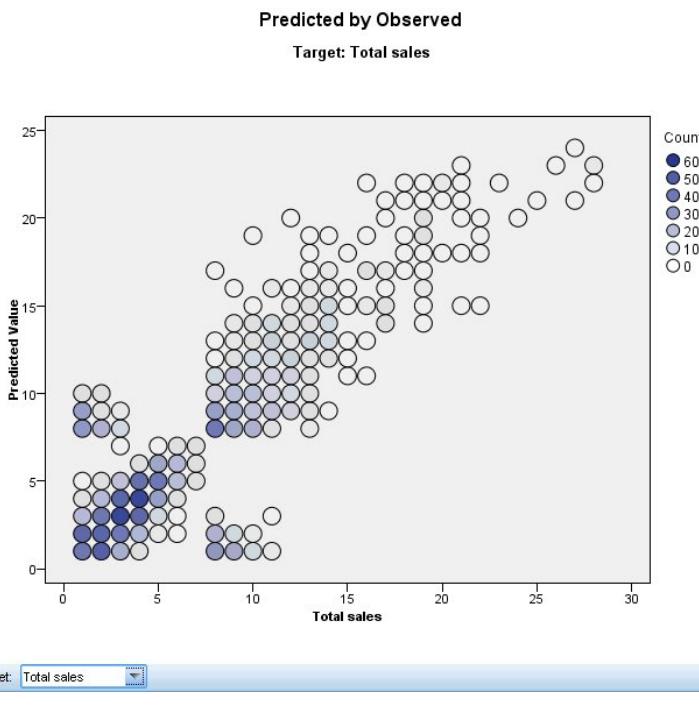


Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

Multiple targets. If there are multiple targets, then each target is displayed in a separate chart and there is a Target drop-down that controls which target to display.

Predicted By Observed (neural networks)

Figure 1. Predicted By Observed view

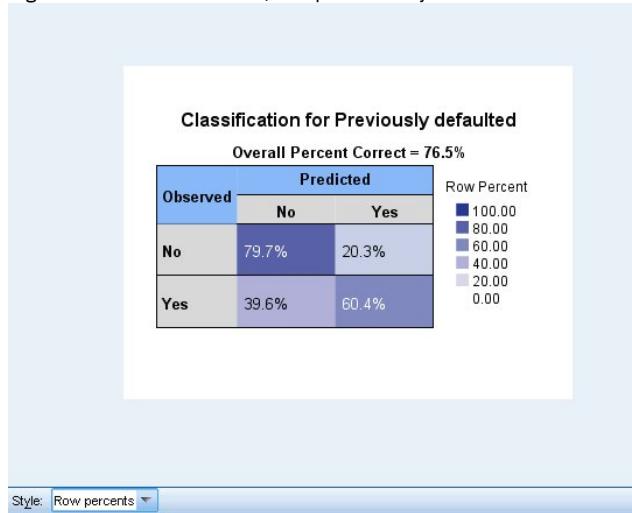


For continuous targets, this displays a binned scatterplot of the predicted values on the vertical axis by the observed values on the horizontal axis.

Multiple targets. If there are multiple continuous targets, then each target is displayed in a separate chart and there is a Target drop-down that controls which target to display.

Classification (neural networks)

Figure 1. Classification view, row percents style



For categorical targets, this displays the cross-classification of observed versus predicted values in a heat map, plus the overall percent correct.

Table styles. There are several different display styles, which are accessible from the Style dropdown list.

- Row percents. This displays the row percentages (the cell counts expressed as a percent of the row totals) in the cells. This is the default.
- Cell counts. This displays the cell counts in the cells. The shading for the heat map is still based on the row percentages.
- Heat map. This displays no values in the cells, just the shading.
- Compressed. This displays no row or column headings, or values in the cells. It can be useful when the target has a lot of categories.

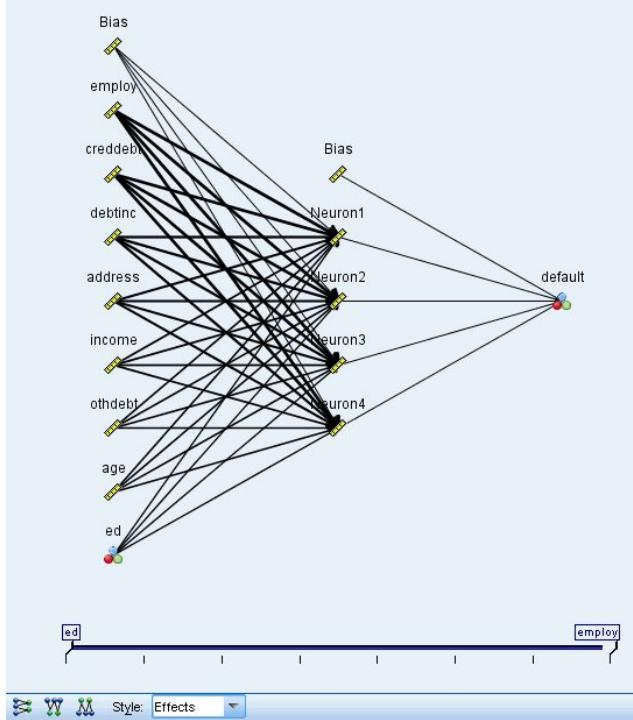
Missing. If any records have missing values on the target, they are displayed in a (Missing) row under all valid rows. Records with missing values do not contribute to the overall percent correct.

Multiple targets. If there are multiple categorical targets, then each target is displayed in a separate table and there is a Target dropdown list that controls which target to display.

Large tables. If the displayed target has more than 100 categories, no table is displayed.

Network (neural networks)

Figure 1. Network view, inputs on the left, effects style



This displays a graphical representation of the neural network.

Chart styles. There are two different display styles, which are accessible from the Style drop-down.

- Effects. This displays each predictor and target as one node in the diagram irrespective of whether the measurement scale is continuous or categorical. This is the default.
- Coefficients. This displays multiple indicator nodes for categorical predictors and targets. The connecting lines in the coefficients-style diagram are colored based on the estimated value of the synaptic weight.

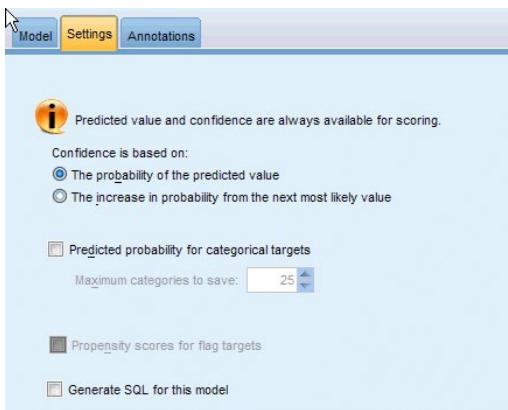
Diagram orientation. By default, the network diagram is arranged with the inputs on the left and the targets on the right. Using toolbar controls, you can change the orientation so that inputs are on top and targets on the bottom, or inputs on the bottom and targets on top.

Predictor importance. Connecting lines in the diagram are weighted based on predictor importance, with greater line width corresponding to greater importance. There is a Predictor Importance slider in the toolbar that controls which predictors are shown in the network diagram. This does not change the model, but simply allows you to focus on the most important predictors.

Multiple targets. If there are multiple targets, all targets are displayed in the chart.

Settings (neural networks)

Figure 1. Settings tab



When the model is scored, the selected items in this tab should be produced. The predicted value (for all targets) and confidence (for categorical targets) are always computed when the model is scored. The computed confidence can be based on the probability of the predicted value (the highest predicted probability) or the difference between the highest predicted probability and the second highest predicted probability.

- Predicted probability for categorical targets. This produces the predicted probabilities for categorical targets. A field is created for each category.
- Propensity scores for flag targets. For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. The model produces raw propensity scores; if partitions are in effect, the model also produces adjusted propensity scores based on the testing partition.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.

Score by converting to native SQL If selected, generates native SQL to score the model within the database.

Note: Although this option can provide quicker results, the size and complexity of the native SQL increases as the complexity of the model increases.

Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Decision List

Decision List models identify subgroups or **segments** that show a higher or lower likelihood of a binary (yes or no) outcome relative to the overall sample. For example, you might look for customers who are least likely to churn or most likely to say yes to a particular offer or campaign. The Decision List Viewer gives you complete control over the model, enabling you to edit segments, add your own business rules, specify how each segment is scored, and customize the model in a number of other ways to optimize the proportion of hits across all segments. As such, it is particularly well-suited for generating mailing lists or otherwise identifying which records to target for a particular campaign. You can also use multiple **mining tasks** to combine modeling approaches—for example, by identifying high- and low-performing segments within the same model and including or excluding each in the scoring stage as appropriate.

Segments, Rules, and Conditions

A model consists of a list of segments, each of which is defined by a rule that selects matching records. A given rule may have multiple conditions; for example:

```
RFM_SCORE > 10 and
MONTHS_CURRENT <= 9
```

Rules are applied in the order listed, with the first matching rule determining the outcome for a given record. Taken independently, rules or conditions may overlap, but the order of rules resolves ambiguity. If no rule matches, the record is assigned to the remainder rule.

Complete Control over Scoring

The Decision List Viewer enables you to view, modify, and reorganize segments and to choose which to include or exclude for purposes of scoring. For example, you can choose to exclude one group of customers from future offers and include others and immediately see how this affects your overall hit rate. Decision List models return a score of Yes for included segments and \$null\$ for everything else, including the remainder. This direct control over scoring makes Decision List models ideal for generating mailing lists, and they are widely used in customer relationship management, including call center or marketing applications.

Mining Tasks, Measures, and Selections

The modeling process is driven by **mining tasks**. Each mining task effectively initiates a new modeling run and returns a new set of alternative models to choose from. The default task is based on your initial specifications in the Decision List node, but you can define any number of custom tasks. You can also apply tasks iteratively—for example, you can run a high probability search on the entire training set and then run a low probability search on the remainder to weed out low-performing segments.

Data Selections

You can define data selections and custom model measures for model building and evaluation. For example, you can specify a data selection in a mining task to tailor the model to a specific region and create a custom measure to evaluate how well that model performs on the whole country. Unlike mining tasks, measures don't change the underlying model but provide another lens to assess how well it performs.

Adding Your Business Knowledge

By fine-tuning or extending the segments identified by the algorithm, the Decision List Viewer enables you to incorporate your business knowledge right into the model. You can edit the segments generated by the model or add additional segments based on rules that you specify. You can then apply the changes and preview the results.

For further insight, a dynamic link with Excel enables you to export your data to Excel, where it can be used to create presentation charts and to calculate custom measures, such as complex profit and ROI, which can be viewed in the Decision List Viewer while you are building the model.

Example. The marketing department of a financial institution wants to achieve more profitable results in future campaigns by matching the right offer to each customer. You can use a Decision List model to identify the characteristics of customers most likely to respond favorably based on previous promotions and to generate a mailing list based on the results.

Requirements. A single categorical target field with a measurement level of type *Flag* or *Nominal* that indicates the binary outcome you want to predict (yes/no), and at least one input field. When the target field type is *Nominal*, you must manually choose a single value to be treated as a **hit**, or **response**; all the other values are lumped together as **not hit**. An optional frequency field may also be specified. Continuous date/time fields are ignored. Continuous numeric range inputs are automatically binned by the algorithm as specified on the Expert tab in the modeling node. For finer control over binning, add an upstream binning node and use the binned field as input with a measurement level of *Ordinal*.

- [Decision List Model Options](#)
- [Decision List Node Expert Options](#)
- [Decision List Model Nugget](#)
- [Decision List Viewer](#)

Related information

- [Decision List Model Options](#)
 - [Decision List Node Expert Options](#)
 - [Decision List Model Nugget](#)
 - [Decision List Model Nugget Settings](#)
-

Decision List Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Create split models. Builds a separate model for each possible value of input fields that are specified as split fields. See [Building Split Models](#) for more information.

Mode. Specifies the method used to build the model.

- **Generate model.** Automatically generates a model on the models palette when the node is executed. The resulting model can be added to streams for purposes of scoring but cannot be further edited.
- **Launch interactive session.** Opens the interactive Decision List Viewer modeling (output) window, enabling you to pick from multiple alternatives and repeatedly apply the algorithm with different settings to progressively grow or modify the model. See the topic [Decision List Viewer](#) for more information.
- **Use saved interactive session information.** Launches an interactive session using previously saved settings. Interactive settings can be saved from the Decision List Viewer using the Generate menu (to create a model or modeling node) or the File menu (to update the node from which the session was launched).

Target value. Specifies the value of the target field that indicates the outcome you want to model. For example, if the target field churn is coded 0 = **no** and 1 = **yes**, specify 1 to identify rules that indicate which records are likely to churn.

Find segments with. Indicates whether the search for the target variable should look for a High probability or Low probability of occurrence. Finding and excluding them can be a useful way to improve your model and can be particularly useful when the remainder has a low probability.

Maximum number of segments. Specifies the maximum number of segments to return. The top N segments are created, where the best segment is the one with the highest probability or, if more than one model has the same probability, the highest coverage. The minimum allowed setting is 1; there is no maximum setting.

Minimum segment size. The two settings below dictate the minimum segment size. The larger of the two values takes precedence. For example, if the percentage value equates to a number higher than the absolute value, the percentage setting takes precedence.

- **As percentage of previous segment (%)**. Specifies the minimum group size as a percentage of records. The minimum allowed setting is 0; the maximum allowed setting is 99.9.
- **As absolute value (N)**. Specifies the minimum group size as an absolute number of records. The minimum allowed setting is 1; there is no maximum setting.

Segment rules.

Maximum number of attributes. Specifies the maximum number of conditions per segment rule. The minimum allowed setting is 1; there is no maximum setting.

- **Allow attribute re-use.** When enabled, each cycle can consider all attributes, even those that have been used in previous cycles. The conditions for a segment are built up in cycles, where each cycle adds a new condition. The number of cycles is defined using the Maximum number of attributes setting.

Confidence interval for new conditions (%). Specifies the confidence level for testing segment significance. This setting plays a significant role in the number of segments (if any) that are returned as well as the number-of-conditions-per-segment rule. The higher the value, the smaller the returned result set. The minimum allowed setting is 50; the maximum allowed setting is 99.9.

Related information

- [Decision List](#)
 - [Decision List Node Expert Options](#)
 - [Decision List Model Nugget](#)
 - [Decision List Model Nugget Settings](#)
-

Decision List Node Expert Options

Expert options enable you to fine-tune the model-building process.

Binning method. The method used for binning continuous fields (equal count or equal width).

Number of bins. The number of bins to create for continuous fields. The minimum allowed setting is 2; there is no maximum setting.

Model search width. The maximum number of model results per cycle that can be used for the next cycle. The minimum allowed setting is 1; there is no maximum setting.

Rule search width. The maximum number of rule results per cycle that can be used for the next cycle. The minimum allowed setting is 1; there is no maximum setting.

Bin merging factor. The minimum amount by which a segment must grow when merged with its neighbor. The minimum allowed setting is 1.01; there is no maximum setting.

- **Allow missing values in conditions.** `True` to allow the `IS MISSING` test in rules.
 - **Discard intermediate results.** When `True`, only the final results of the search process are returned. A final result is a result that is not refined any further in the search process. When `False`, intermediate results are also returned.
- Maximum number of alternatives.** Specifies the maximum number of alternatives that can be returned upon running the mining task. The minimum allowed setting is 1; there is no maximum setting.

Note that the mining task will only return the actual number of alternatives, up to the maximum specified. For example, if the maximum is set to 100 and only 3 alternatives are found, only those 3 are shown.

Related information

- [Decision List](#)
 - [Decision List Model Options](#)
 - [Decision List Model Nugget](#)
 - [Decision List Model Nugget Settings](#)
-

Decision List Model Nugget

A model consists of a list of **segments**, each of which is defined by a **rule** that selects matching records. You can easily view or modify the segments before generating the model and choose which ones to include or exclude. When used in scoring, Decision List models return Yes for included segments and \$null\$ for everything else, including the remainder. This direct control over scoring makes Decision List models ideal for generating mailing lists, and they are widely used in customer relationship management, including call center or marketing applications.

When you run a stream containing a Decision List model, the node adds three new fields containing the score, either 1 (meaning Yes) for included fields or \$null\$ for excluded fields, the probability (hit rate) for the segment within which the record falls, and the ID number for the segment. The names of the new fields are derived from the name of the output field being predicted, prefixed with \$D- for the score, \$DP- for the probability, and \$DI- for the segment ID.

The model is scored based on the target value specified when the model was built. You can manually exclude segments so that they score as \$null\$. For example, if you run a low probability search to find segments with lower than average hit rates, these “low” segments will be scored as Yes unless you manually exclude them. If necessary, nulls can be recoded as No using a Derive or Filler node.

PMMI

A Decision List model can be stored as a PMML RuleSetModel with a “first hit” selection criterion. However, all of the rules are expected to have the same score. To allow for changes to the target field or the target value, multiple rule set models can be stored in one file to be applied in order, cases not matched by the first model being passed to the second, and so on. The algorithm name *DecisionList* is used to indicate this non-standard behavior, and only rule set models with this name are recognized as Decision List models and scored as such.

- [Decision List Model Nugget Settings](#)

Related information

- [Decision List](#)
 - [Decision List Model Options](#)
 - [Decision List Node Expert Options](#)
 - [Decision List Model Nugget Settings](#)
-

Decision List Model Nugget Settings

The Settings tab for a Decision List model nugget enables you to obtain propensity scores and to enable or disable SQL optimization. This tab is available only after adding the model nugget to a stream.

Calculate raw propensity scores. For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

Calculate adjusted propensity scores. Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score by converting to native SQL If selected, generates native SQL to score the model within the database.
Note: Although this option can provide quicker results, the size and complexity of the native SQL increases as the complexity of the model increases.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Related information

- [Decision List](#)
 - [Decision List Model Options](#)
 - [Decision List Node Expert Options](#)
 - [Decision List Model Nugget](#)
-

Decision List Viewer

The easy-to-use, task-based Decision List Viewer graphical interface takes the complexity out of the model building process, freeing you from the low-level details of data mining techniques and enabling you to devote your full attention to those parts of the analysis requiring user intervention, such as setting objectives, choosing target groups, analyzing the results, and selecting the optimal model.

- [Working Model Pane](#)
- [Alternatives Tab](#)
- [Snapshots Tab](#)
- [Working with Decision List Viewer](#)

Related information

- [Working with Decision List Viewer](#)
 - [Working Model Pane](#)
 - [Alternatives Tab](#)
 - [Snapshots Tab](#)
-

Working Model Pane

The working model pane displays the current model, including mining tasks and other actions that apply to the working model.

ID. Identifies the sequential segment order. Model segments are calculated, in sequence, according to their ID number.

Segment Rules. Provides the segment name and defined segment conditions. By default, the segment name is the field name or concatenated field names used in the conditions, with a comma as a separator.

Score. Represents the field that you want to predict, whose value is assumed to be related to the values of other fields (the predictors).

Note: The following options can be toggled to display via the [Organizing Model Measures](#) dialog.

Cover. The pie chart visually identifies the coverage each segment has in relation to the entire cover.

Cover (n). Lists the coverage for each segment in relation to the entire cover.

Frequency. Lists the number of hits received in relation to the cover. For example, when the cover is 79 and the frequency is 50, that means that 50 out of 79 responded for the selected segment.

Probability. Indicates the segment probability. For example, when the cover is 79 and the frequency is 50, that means that the probability for the segment is 63.29% (50 divided by 79).

Error. Indicates the segment error.

The information at the bottom of the pane indicates the cover, frequency, and probability for the entire model.

Working Model Toolbar

The working model pane provides the following functions via a toolbar.

Note: Some functions are also available by right-clicking a model segment.

Table 1. Working model toolbar buttons

Toolbar Button	Description
	Launches the Generate New Model dialog, which provides options for creating a new model nugget.
	Saves the current state of the interactive session. The Decision List modeling node is updated with the current settings, including mining tasks, model snapshots, data selections, and custom measures. To restore a session to this state, check the Use saved session information box on the Model tab of the modeling node and click Run.
	Displays the Organize Model Measures dialog. See the topic Organizing Model Measures for more information.
	Displays the Organize Data Selections dialog. See the topic Organizing Data Selections for more information.
	Displays the Snapshots tab. See the topic Snapshots Tab for more information.
	Displays the Alternatives tab. See the topic Alternatives Tab for more information.
	Takes a snapshot of the current model structure. Snapshots display on the Snapshots tab and are commonly used for model comparison purposes.
	Launches the Inserting Segments dialog, which provides options for creating new model segments.
	Launches the Editing Segment Rules dialog, which provides options for adding conditions to model segments or changing previously defined model segment conditions.

	Moves the selected segment up in the model hierarchy.
	Moves the selected segment down in the model hierarchy.
	Deletes the selected segment.
	Toggles whether the selected segment is included in the model. When excluded, the segment results are added to the remainder. This differs from deleting a segment in that you have the option of reactivating the segment.

Related information

- [Decision List Viewer](#)
 - [Alternatives Tab](#)
 - [Snapshots Tab](#)
-

Alternatives Tab

Generated when you click Find Segments, the Alternatives tab lists all alternative mining results for the selected model or segment on the working model pane.

To promote an alternative to be the working model, highlight the required alternative and click Load; the alternative model is displayed in the working model pane.

Note: The Alternatives tab is only displayed if you have set Maximum number of alternatives on the Decision List modeling node Expert tab to create more than one alternative.

Each generated model alternative displays specific model information:

Name. Each alternative is sequentially numbered. The first alternative usually contains the best results.

Target. Indicates the target value. For example: 1, which equals "true".

No. of Segments. The number of segment rules used in the alternative model.

Cover. The coverage of the alternative model.

Freq. The number of hits in relation to the cover.

Prob. Indicates the probability percentage of the alternative model.

Note: Alternative results are not saved with the model; results are valid only during the active session.

Related information

- [Decision List Viewer](#)
 - [Working Model Pane](#)
 - [Snapshots Tab](#)
-

Snapshots Tab

A snapshot is a view of a model at a specific point in time. For example, you could take a model snapshot when you want to load a different alternative model into the working model pane but do not want to lose the work on the current model. The Snapshots tab lists all model snapshots manually taken for any number of working model states.

Note: Snapshots are saved with the model. We recommend that you take a snapshot when you load the first model. This snapshot will then preserve the original model structure, ensuring that you can always return to the original model state. The generated snapshot name displays as a timestamp, indicating when it was generated.

Create a Model Snapshot

1. Select an appropriate model/alternative to display in the working model pane.
2. Make any necessary changes to the working model.
3. Click Take Snapshot. A new snapshot is displayed on the Snapshots tab.

Name. The snapshot name. You can change a snapshot name by double-clicking the snapshot name.

Target. Indicates the target value. For example: 1, which equals "true".

No. of Segments. The number of segment rules used in the model.

Cover. The coverage of the model.

Freq. The number of hits in relation to the cover.

Prob. Indicates the probability percentage of the model.

4. To promote a snapshot to be the working model, highlight the required snapshot and click Load; the snapshot model is displayed in the working model pane.
5. You can delete a snapshot by clicking Delete or by right-clicking the snapshot and choosing Delete from the menu.

Related information

- [Decision List Viewer](#)
 - [Working Model Pane](#)
 - [Alternatives Tab](#)
-

Working with Decision List Viewer

A model that will best predict customer response and behavior is built in various stages. When Decision List Viewer launches, the working model is populated with the defined model segments and measures, ready for you start a mining task, modify the segments/measures as required, and generate a new model or modeling node.

You can add one or more segment rules until you have developed a satisfactory model. You can add segment rules to the model by running mining tasks or by using the Edit Segment Rule function.

In the model building process, you can assess the performance of the model by validating the model against measure data, by visualizing the model in a chart, or by generating custom Excel measures.

When you feel certain about the model's quality, you can generate a new model and place it on the IBM® SPSS® Modeler canvas or Model palette.

- [Mining Tasks](#)
- [Undo and Redo Actions](#)
- [Segment Rules](#)
- [Customizing a Model](#)
- [Generate New Model](#)
- [Model Assessment](#)
- [Visualizing Models](#)

Related information

- [Decision List Viewer](#)
 - [Undo and Redo Actions](#)
 - [Segment Rules](#)
 - [Customizing a Model](#)
 - [Generate New Model](#)
 - [Model Assessment](#)
 - [Assessment in Excel](#)
 - [Visualizing Models](#)
-

Mining Tasks

A **mining task** is a collection of parameters that determines the way new rules are generated. Some of these parameters are selectable to provide you with the flexibility to adapt models to new situations. A task consists of a task template (type), a target, and a build selection (mining dataset).

The following sections detail the various mining task operations:

- [Running Mining Tasks](#)
- [Creating and Editing a Mining Task](#)
- [Organizing Data Selections](#)
- [Running Mining Tasks](#)

- [Creating and Editing a Mining Task](#)
- [Organizing Data Selections](#)

Related information

- [Running Mining Tasks](#)
- [Creating and Editing a Mining Task](#)
- [New Settings](#)
- [Organizing Data Selections](#)
- [Specify Selection Condition](#)
- [Insert Value](#)
- [Segment Rules](#)
- [Inserting Segments](#)
- [Editing Segment Rules](#)
- [Copying Segments](#)
- [Alternative Models](#)

Running Mining Tasks

Decision List Viewer enables you to manually add segment rules to a model by running mining tasks or by copying and pasting segment rules between models. A mining task holds information on how to generate new segment rules (the data mining parameter settings, such as the search strategy, source attributes, search width, confidence level, and so on), the customer behavior to predict, and the data to investigate. The goal of a mining task is to search for the best possible segment rules.

To generate a model segment rule by running a mining task:

1. Click the Remainder row. If there are already segments displayed on the working model pane, you can also select one of the segments to find additional rules based on the selected segment. After selecting the remainder or segment, use one of the following methods to generate the model, or alternative models:
 - From the Tools menu, choose Find Segments.
 - Right-click the Remainder row/segment and choose Find Segments.
 - Click the Find Segments button on the working model pane.

While the task is processing, the progress is displayed at the bottom of the workspace and informs you when the task has completed. Precisely how long a task takes to complete depends on the complexity of the mining task and the size of the dataset. If there is only a single model in the results it is displayed on the working model pane as soon as the task completes; however, where the results contain more than one model they are displayed on the Alternatives tab.

Note: A task result will either complete with models, complete with no models, or fail.

The process of finding new segment rules can be repeated until no new rules are added to the model. This means that all significant groups of customers have been found.

It is possible to run a mining task on any existing model segment. If the result of a task is not what you are looking for, you can choose to start another mining task on the same segment. This will provide additional found rules based on the selected segment. Segments that are "below" the selected segment (that is, added to the model later than the selected segment) are replaced by the new segments because each segment depends on its predecessors.

Related information

- [Mining Tasks](#)
- [Creating and Editing a Mining Task](#)
- [New Settings](#)
- [Organizing Data Selections](#)
- [Specify Selection Condition](#)
- [Insert Value](#)

Creating and Editing a Mining Task

A mining task is the mechanism that searches for the collection of rules that make up a data model. Alongside the search criteria defined in the selected template, a task also defines the target (the actual question that motivated the analysis, such as how many customers are likely to respond to a mailing), and it identifies the datasets to be used. The goal of a mining task is to search for the best possible models.

Create a mining task

To create a mining task:

1. Select the segment from which you want to mine additional segment conditions.
2. Click Settings. The Create/Edit Mining Task dialog opens. This dialog provides options for defining the mining task.
3. Make any necessary changes and click OK to return to the working model pane. Decision List Viewer uses the settings as the defaults to run for each task until an alternative task or settings is selected.
4. Click Find Segments to start the mining task on the selected segment.

Edit a mining task

The Create/Edit Mining Task dialog provides options for defining a new mining task or editing an existing one.

Most parameters available for mining tasks are similar to those offered in the Decision List node. The exceptions are shown below. See the topic [Decision List Model Options](#) for more information.

Load Settings: When you have created more than one mining task, select the required task.

New... Click to create a new mining task based on the settings of the task currently displayed.

See the topic [New Settings](#) for more information.

Target

Target Field: Represents the field that you want to predict, whose value is assumed to be related to the values of other fields (the predictors).

Target value. Specifies the value of the target field that indicates the outcome you want to model. For example, if the target field churn is coded **0 = no** and **1 = yes**, specify **1** to identify rules that indicate which records are likely to churn.

Simple Settings

Maximum number of alternatives. Specifies the number of alternatives that will be displayed upon running the mining task. The minimum allowed setting is **1**; there is no maximum setting.

Expert Settings

Edit... Opens the Edit Advanced Parameters dialog that enables you to define the advanced settings. See the topic [Edit Advanced Parameters](#) for more information.

Data

Build selection. Provides options for specifying the evaluation measure that Decision List Viewer should analyze to find new rules. The listed evaluation measures are created/edited in the Organize Data Selections dialog.

Available fields. Provides options for displaying all fields or manually selecting which fields to display.

Edit... If the Custom option is selected, this opens the Customize Available Fields dialog that enables you to select which fields are available as segment attributes found by the mining task. See the topic [Customize Available Fields](#) for more information.

- [New Settings](#)
- [Edit Advanced Parameters](#)
- [Customize Available Fields](#)

Related information

- [Mining Tasks](#)
 - [Running Mining Tasks](#)
 - [New Settings](#)
 - [Organizing Data Selections](#)
 - [Specify Selection Condition](#)
 - [Insert Value](#)
-

New Settings

When you click **New...** on the Create/Edit Mining Task dialog, the New Settings dialog is displayed.

Enter an appropriate name and click **OK**. The Create/Edit Mining Task dialog is displayed ready for you to amend the settings.

Related information

- [Mining Tasks](#)
 - [Running Mining Tasks](#)
 - [Creating and Editing a Mining Task](#)
 - [Organizing Data Selections](#)
 - [Specify Selection Condition](#)
 - [Insert Value](#)
-

Edit Advanced Parameters

The Edit Advanced Parameters dialog provides the following configuration options.

Binning method. The method used for binning continuous fields (equal count or equal width).

Number of bins. The number of bins to create for continuous fields. The minimum allowed setting is 2; there is no maximum setting.

Model search width. The maximum number of model results per cycle that can be used for the next cycle. The minimum allowed setting is 1; there is no maximum setting.

Rule search width. The maximum number of rule results per cycle that can be used for the next cycle. The minimum allowed setting is 1; there is no maximum setting.

Bin merging factor. The minimum amount by which a segment must grow when merged with its neighbor. The minimum allowed setting is 1.01; there is no maximum setting.

- **Allow missing values in conditions.** **True** to allow the `IS MISSING` test in rules.
 - **Discard intermediate results.** When **True**, only the final results of the search process are returned. A final result is a result that is not refined any further in the search process. When **False**, intermediate results are also returned.
-

Customize Available Fields

The Customize Available Fields dialog enables you to select which fields are available as segment attributes found by the mining task.

Available. Lists the fields that are currently available as segment attributes. To remove fields from the list, select the appropriate fields and click Remove >>. The selected fields move from the Available list to the Not Available list.

Not Available. Lists the fields that are not available as segment attributes. To include the fields in the available list, select the appropriate fields and click << Add. The selected fields move from the Not Available list to the Available list.

Organizing Data Selections

By organizing data selections (a mining dataset), you can specify which evaluation measures Decision List Viewer should analyze to find new rules and select which data selections are used as the basis for measures.

To organize data selections:

1. From the Tools menu, choose Organize Data Selections, or right-click a segment and choose the option. The Organize Data Selections dialog opens.
Note: The Organize Data Selections dialog also enables you to edit or delete existing data selections.
2. Click the Add new data selection button. A new data selection entry is added to the existing table.
3. Click Name and enter an appropriate selection name.
4. Click Partition and select an appropriate partition type.
5. Click Condition and select an appropriate condition option. When Specify is selected, the Specify Selection Condition dialog opens, providing options for defining specific field conditions.
6. Define the appropriate condition and click OK.

The data selections are available from the Build Selection drop-down list in the Create/Edit Mining Task dialog. The list enables you to select which evaluation measure is used for a particular mining task.

- [Specify Selection Condition](#)

Related information

- [Mining Tasks](#)
 - [Running Mining Tasks](#)
 - [Creating and Editing a Mining Task](#)
 - [New Settings](#)
 - [Specify Selection Condition](#)
 - [Insert Value](#)
-

Specify Selection Condition

The Specify Selection Condition dialog provides options for defining selection conditions for data selection partitions. The conditions are used to select records from the defined partition. For example, the default condition for the **Training** partition is **All Records**. By specifying a selection condition for a new or existing data selection, you could filter **All Records** down into more concise results such as `married=yes`. When the data selection is used in a mining task, the result set will only return hits for individuals who are married.

Attribute – lists the available input fields.

Operator – the drop-down list provides all valid operators for the selected input field.

Values – queries values for the selected input field. For example the input field `married` would provide the options YES and NO.

- [Insert Value](#)

Related information

- [Mining Tasks](#)
 - [Running Mining Tasks](#)
 - [Creating and Editing a Mining Task](#)
 - [New Settings](#)
 - [Organizing Data Selections](#)
 - [Insert Value](#)
-

Insert Value

The Insert Value dialog displays the available values for the selected input field.

Select an appropriate value and click Insert

Related information

- [Mining Tasks](#)
 - [Running Mining Tasks](#)
 - [Creating and Editing a Mining Task](#)
 - [New Settings](#)
 - [Organizing Data Selections](#)
 - [Specify Selection Condition](#)
-

Undo and Redo Actions

Decision List Viewer enables you to undo and redo the last 10 actions that you have performed. For example you can:

- Undo new model rules
- Undo changes you have made to a model
- Undelete elements of a model segment
- Redo selecting alternative models.

To undo or redo an action, from the Edit menu, choose either Undo or Redo.

Related information

- [Working with Decision List Viewer](#)
 - [Segment Rules](#)
 - [Customizing a Model](#)
 - [Generate New Model](#)
 - [Model Assessment](#)
 - [Assessment in Excel](#)
 - [Visualizing Models](#)
-

Segment Rules

You find model segment rules by running a mining task based on a task template. You can manually add segment rules to a model using the Insert Segment or Edit Segment Rule functions.

If you choose to mine for new segment rules, the results, if any, are displayed on the Viewer tab of the Interactive List dialog. You can quickly refine your model by selecting one of the alternative results from the Model Albums dialog and clicking Load. In this way, you can experiment with differing results until you are ready to build a model that accurately describes your optimum target group.

- [Inserting Segments](#)
- [Editing Segment Rules](#)
- [Copying Segments](#)
- [Alternative Models](#)

Related information

- [Mining Tasks](#)
 - [Inserting Segments](#)
 - [Editing Segment Rules](#)
 - [Copying Segments](#)
 - [Alternative Models](#)
 - [Working with Decision List Viewer](#)
 - [Undo and Redo Actions](#)
 - [Customizing a Model](#)
 - [Generate New Model](#)
 - [Model Assessment](#)
 - [Assessment in Excel](#)
 - [Visualizing Models](#)
-

Inserting Segments

You can manually add segment rules to a model using the Insert Segment function.

To add a segment rule condition to a model:

1. In the Interactive List dialog, select a location where you want to add a new segment. The new segment will be inserted directly above the selected segment.
2. From the Edit menu, choose Insert Segment, or access this selection by right-clicking a segment.
The Insert Segment dialog opens, enabling you to insert new segment rule conditions.
3. Click Insert. The Insert Condition dialog opens, enabling you to define the attributes for the new rule condition.
4. Select a field and an operator from the drop-down lists.
Note: If you select the Not in operator, the selected condition will function as an exclusion condition and displays in red in the Insert Rule dialog. For example, when the condition `region = 'TOWN'` displays in red, it means that `TOWN` is excluded from the result set.
5. Enter one or more values or click the Insert Value icon to display the Insert Value dialog. The dialog enables you to choose a value defined for the selected field. For example, the field married will provide the values yes and no.
6. Click OK to return to the Insert Segment dialog. Click OK a second time to add the created segment to the model.

The new segment will display in the specified model location.

Related information

- [Mining Tasks](#)
 - [Segment Rules](#)
 - [Editing Segment Rules](#)
 - [Copying Segments](#)
 - [Alternative Models](#)
-

Editing Segment Rules

The Edit Segment Rule functionality enables you to add, change, or delete segment rule conditions.

To change a segment rule condition:

1. Select the model segment that you want to edit.
2. From the Edit menu, choose Edit Segment Rule, or right-click on the rule to access this selection.
The Edit Segment Rule dialog opens.
3. Select the appropriate condition and click Edit.
The Edit Condition dialog opens, enabling you to define the attributes for the selected rule condition.
4. Select a field and an operator from the drop-down lists.
Note: If you select the Not in operator, the selected condition will function as an exclusion condition and displays in red in the Edit Segment Rule dialog. For example, when the condition `region = 'TOWN'` displays in red, it means that `TOWN` is excluded from the result set.
5. Enter one or more values or click the Insert Value button to display the Insert Value dialog. The dialog enables you to choose a value defined for the selected field. For example, the field married will provide the values yes and no.
6. Click OK to return to the Edit Segment Rule dialog. Click OK a second time to return to the working model.

The selected segment will display with the updated rule conditions.

- [Insert/Edit Condition](#)
 - [Deleting Segment Rule Conditions](#)
-

Related information

- [Insert/Edit Condition](#)
 - [Deleting Segment Rule Conditions](#)
 - [Mining Tasks](#)
 - [Segment Rules](#)
 - [Inserting Segments](#)
 - [Copying Segments](#)
 - [Alternative Models](#)
-

Insert/Edit Condition

The Insert/Edit Condition dialog provides options for defining segment rule conditions.

Attribute – lists the available input fields.

Operator – the drop-down provides all valid operators for the selected input field.

Values – queries values for the selected input field. For example the input field married would provide the value options YES and NO.

Note: If you select the **Not in** operator, the selected condition will function as an exclusion condition and displays in red in the **Insert/Edit Segment** dialog. For example when the condition `region = 'TOWN'` displays is red it means that `TOWN` is excluded from the result set.

Related information

- [Editing Segment Rules](#)
 - [Deleting Segment Rule Conditions](#)
-

Deleting Segment Rule Conditions

To delete a segment rule condition:

1. Select the model segment containing the rule conditions that you want to delete.
2. From the Edit menu, choose Edit Segment Rule, or right-click on the segment to access this selection. The Edit Segment Rule dialog opens, enabling you to delete one or more segment rule conditions.
3. Select the appropriate rule condition and click Delete.
4. Click OK.

Deleting one or more segment rule conditions causes the working model pane to refresh its measure metrics.

Related information

- [Editing Segment Rules](#)
 - [Insert/Edit Condition](#)
-

Copying Segments

Decision List Viewer provides you with a convenient way to copy model segments. When you want to apply a segment from one model to another model, simply copy (or cut) the segment from one model and paste it into another model. You can also copy a segment from a model displayed in the Alternative Preview panel and paste it into the model displayed in the working model pane. These cut, copy, and paste functions use a system clipboard to store or retrieve temporary data. This means that in the clipboard the conditions and target are copied. The clipboard contents are not solely reserved to be used in Decision List Viewer but can also be pasted in other applications. For example, when the clipboard contents are pasted in a text editor, the conditions and target are pasted in XML-format.

To copy or cut model segments:

1. Select the model segment that you want to use in another model.
2. From the Edit menu, choose **Copy** (or **Cut**), or right-click on the model segment and select **Copy** or **Cut**.
3. Open the appropriate model (where the model segment will be pasted).
4. Select one of the model segments, and click **Paste**.

Note: Instead of the **Cut**, **Copy**, and **Paste** commands you can also use the key combinations: **Ctrl+X**, **Ctrl+C**, and **Ctrl+V**.

The copied (or cut) segment is inserted above the previously selected model segment. The measures of the pasted segment and segments below are recalculated.

Note: Both models in this procedure must be based on the same underlying model template and contain the same target, otherwise an error message is displayed.

Related information

- [Mining Tasks](#)
 - [Segment Rules](#)
 - [Inserting Segments](#)
 - [Editing Segment Rules](#)
 - [Alternative Models](#)
-

Alternative Models

Where there is more than one result, the Alternatives tab displays the results of each mining task. Each result consists of the conditions in the selected data that most closely match the target, as well as any "good enough" alternatives. The total number of alternatives shown depends on the search criteria used in the analysis process.

To view alternative models:

1. Click on an alternative model on the Alternatives tab. The alternative model segments display, or replace the current model segments, in the Alternative Preview panel.
2. To work with an alternative model in the working model pane, select the model and click Load in the Alternative Preview panel or right-click an alternative name on the Alternatives tab and choose Load.

Note: Alternative models are not saved when you generate a new model.

Related information

-
- [Mining Tasks](#)
 - [Segment Rules](#)
 - [Inserting Segments](#)
 - [Editing Segment Rules](#)
 - [Copying Segments](#)
-

Customizing a Model

Data are not static. Customers move, get married, and change jobs. Products lose market focus and become obsolete.

Decision List Viewer offers business users the flexibility to adapt models to new situations easily and quickly. You can change a model by editing, prioritizing, deleting, or inactivating specific model segments.

- [Prioritizing Segments](#)
- [Deleting Segments](#)
- [Excluding Segments](#)
- [Change Target Value](#)

Related information

- [Prioritizing Segments](#)
 - [Deleting Segments](#)
 - [Excluding Segments](#)
 - [Working with Decision List Viewer](#)
 - [Undo and Redo Actions](#)
 - [Segment Rules](#)
 - [Generate New Model](#)
 - [Model Assessment](#)
 - [Assessment in Excel](#)
 - [Visualizing Models](#)
-

Prioritizing Segments

You can rank model rules in any order you choose. By default, model segments are displayed in order of priority, the first segment having the highest priority. When you assign a different priority to one or more of the segments, the model is changed accordingly. You may alter the model as required by moving segments to a higher or lower priority position.

To prioritize model segments:

1. Select the model segment to which you want to assign a different priority.
2. Click one of the two arrow buttons on the working model pane toolbar to move the selected model segment up or down the list.

After prioritization, all previous assessment results are recalculated and the new values are displayed.

Related information

- [Customizing a Model](#)
 - [Deleting Segments](#)
 - [Excluding Segments](#)
-

Deleting Segments

To delete one or more segments:

1. Select a model segment.
2. From the Edit menu, choose Delete Segment, or click the delete button on the toolbar of the working model pane.

The measures are recalculated for the modified model, and the model is changed accordingly.

Related information

- [Customizing a Model](#)
 - [Prioritizing Segments](#)
 - [Excluding Segments](#)
-

Excluding Segments

As you are searching for particular groups, you will probably base business actions on a selection of the model segments. When deploying a model, you may choose to exclude segments within a model. Excluded segments are scored as null values. Excluding a segment does not mean the segment is not used; it means that all records matching this rule are excluded from the mailing list. The rule is still applied but differently.

To exclude specific model segments:

1. Select a segment from the working model pane.
2. Click the Toggle Segment Exclusion button on the toolbar of the working model pane. Excluded is now displayed in the selected Target column of the selected segment.

Note: Unlike deleted segments, excluded segments remain available for reuse in the final model. Excluded segments affect chart results.

Related information

- [Customizing a Model](#)
 - [Prioritizing Segments](#)
 - [Deleting Segments](#)
-

Change Target Value

The Change Target Value dialog enables you to change the target value for the current target field.

Snapshots and session results with a different target value than the Working Model are identified by changing the table background for that row to yellow. This indicates that snapshot/session result is outdated.

The **Create/Edit Mining Task** dialog displays the target value for the current working model. The target value is not saved with the mining task. It is instead taken from the Working Model value.

When you promote a saved model to the Working Model that has a different target value from the current working model (for example, by editing an alternative result or editing a copy of a snapshot), the target value of the saved model is changed to be the same as the working model (the target value shown in the Working Model pane is not changed). The model metrics are reevaluated with the new target.

Generate New Model

The Generate New Model dialog provides options for naming the model and selecting where the new node is created.

Model name. Select Custom to adjust the auto-generated name or to create a unique name for the node as displayed on the stream canvas.

Create node on. Selecting Canvas places the new model on the working canvas; selecting GM Palette places the new model on the Models palette; selecting Both places the new model on both the working canvas and the Models palette.

Include interactive session state. When enabled, the interactive session state is preserved in the generated model. When you later generate a modeling node from the model, the state is carried over and used to initialize the interactive session. Regardless of whether the option is selected, the model itself scores new data identically. When the option is not selected, the model is still able to create a build node, but it will be a more generic build node that starts a new interactive session rather than pick up where the old session left off. If you change the node settings but execute with a saved state, the settings you have changed are ignored in favor of the settings from the saved state.

Note: The standard metrics are the only metrics that remain with the model. Additional metrics are preserved with the interactive state. The generated model does not represent the saved interactive mining task state. Once you launch the Decision List Viewer, it displays the settings originally made through the Viewer.

See the topic [Regenerating a modeling node](#) for more information.

Related information

- [Working with Decision List Viewer](#)
 - [Undo and Redo Actions](#)
 - [Segment Rules](#)
 - [Customizing a Model](#)
 - [Model Assessment](#)
 - [Assessment in Excel](#)
 - [Visualizing Models](#)
-

Model Assessment

Successful modeling requires the careful assessment of the model before implementation in the production environment takes place. Decision List Viewer provides a number of statistical and business measures that can be used to assess the impact of a model in the real world. These include gains charts and full interoperability with Excel, thus enabling cost/benefit scenarios to be simulated for assessing the impact of deployment.

You can assess your model in the following ways:

- Using the predefined statistical and business model measures available in Decision List Viewer (probability, frequency).
 - Evaluating measures imported from Microsoft Excel.
 - Visualizing the model using a gains chart.
- [Organizing Model Measures](#)
 - [Assessment in Excel](#)

Related information

- [Organizing Model Measures](#)
 - [Refreshing Measures](#)
 - [Working with Decision List Viewer](#)
 - [Undo and Redo Actions](#)
 - [Segment Rules](#)
 - [Customizing a Model](#)
 - [Generate New Model](#)
 - [Assessment in Excel](#)
 - [Visualizing Models](#)
-

Organizing Model Measures

Decision List Viewer provides options for defining the measures that are calculated and displayed as columns. Each segment can include the default cover, frequency, probability, and error measures represented as columns. You can also create new measures that will be displayed as columns.

Defining Model Measures

To add a measure to your model or to define an existing measure:

1. From the Tools menu, choose Organize Model Measures, or right-click the model to make this selection. The Organize Model Measures dialog opens.
2. Click the Add new model measure button (to the right of the Show column). A new measure is displayed in the table.
3. Provide a measure name and select an appropriate type, display option, and selection. The Show column indicates whether the measure will display for the working model. When defining an existing measure, select an appropriate metric and selection and indicate if the measure will display for the working model.
4. Click OK to return to the Decision List Viewer workspace. If the Show column for the new measure was checked, the new measure will display for the working model.

Custom Metrics in Excel

See the topic [Assessment in Excel](#) for more information.

- [Refreshing Measures](#)

Related information

- [Model Assessment](#)
 - [Refreshing Measures](#)
-

Refreshing Measures

In certain cases, it may be necessary to recalculate the model measures, such as when you apply an existing model to a new set of customers.

To recalculate (refresh) the model measures:

From the Edit menu, choose Refresh All Measures.

or

Press F5.

All measures are recalculated, and the new values are shown for the working model.

Related information

- [Model Assessment](#)
 - [Organizing Model Measures](#)
-

Assessment in Excel

Decision List Viewer can be integrated with Microsoft Excel, enabling you to use your own value calculations and profit formulas directly within the model building process to simulate cost/benefit scenarios. The link with Excel enables you to export data to Excel, where it can be used to create presentation charts, calculate custom measures, such as complex profit and ROI measures, and view them in Decision List Viewer while building the model.

Note: In order for you to work with an Excel spreadsheet, the analytical CRM expert has to define configuration information for the synchronization of Decision List Viewer with Microsoft Excel. The configuration is contained in an Excel spreadsheet file and indicates which information is transferred from Decision List Viewer to Excel, and vice versa.

The following steps are valid only when MS Excel is installed. If Excel is not installed, the options for synchronizing models with Excel are not displayed.

To synchronize models with MS Excel:

1. Open the model, run an interactive session, and choose Organize Model Measures from the Tools menu.
2. Select Yes for the Calculate custom measures in Excel option. The Workbook field activates, enabling you to select a preconfigured Excel workbook template.
3. Click the Connect to Excel button. The Open dialog opens, enabling you to navigate to the preconfigured template location on your local or network file system.
4. Select the appropriate Excel template and click Open. The selected Excel template launches; use the Windows taskbar (or press Alt-Tab) to navigate back to the Choose Inputs for Custom Measures dialog.
5. Select the appropriate mappings between the metric names defined in the Excel template and the model metric names and click OK.

Once the link is established, Excel starts with the preconfigured Excel template that displays the model rules in the spreadsheet. The results calculated in Excel are displayed as new columns in Decision List Viewer.

Note: Excel metrics do not remain when the model is saved; the metrics are valid only during the active session. However, you can create snapshots that include Excel metrics. The Excel metrics saved in the snapshot views are valid only for historical comparison purposes and do not refresh when reopened. See the topic [Snapshots Tab](#) for more information. The Excel metrics will not display in the snapshots until you reestablish a connection to the Excel template.

- [Choose Inputs for Custom Measures](#)
- [MS Excel Integration Setup](#)
- [Changing the Model Measures](#)

Related information

- [Choose Inputs for Custom Measures](#)

- [MS Excel Integration Setup](#)
 - [Working with Decision List Viewer](#)
 - [Undo and Redo Actions](#)
 - [Segment Rules](#)
 - [Customizing a Model](#)
 - [Generate New Model](#)
 - [Model Assessment](#)
 - [Visualizing Models](#)
-

Choose Inputs for Custom Measures

The Choose Inputs for Custom Measures dialog provides options for mapping custom Excel input metrics to existing model measures.

Input – lists the custom Excel input metrics from the Excel template.

Model Measure – lists the available model measures. Select an appropriate measure to map to each custom Excel input metric.

Related information

- [Assessment in Excel](#)
 - [MS Excel Integration Setup](#)
-

MS Excel Integration Setup

The integration between Decision List Viewer and Microsoft Excel is accomplished through the use of a preconfigured Excel spreadsheet template. The template consists of three worksheets:

Model Measures. Displays the imported Decision List Viewer measures, the custom Excel measures, and the calculation totals (defined on the Settings worksheet).

Settings. Provides the variables to generate calculations based on the imported Decision List Viewer measures and the custom Excel measures.

Configuration. Provides options for specifying which measures are imported from Decision List Viewer and for defining the custom Excel measures.

WARNING: The structure of the Configuration worksheet is rigidly defined. Do **NOT** edit any cells in the green shaded area.

- **Metrics From Model.** Indicates which Decision List Viewer metrics are used in the calculations.
- **Metrics To Model.** Indicates which Excel-generated metric(s) will be returned to Decision List Viewer. The Excel-generated metrics display as new measure columns in Decision List Viewer.

Note: Excel metrics do not remain with the model when you generate a new model; the metrics are valid only during the active session.

Related information

- [Assessment in Excel](#)
 - [Choose Inputs for Custom Measures](#)
-

Changing the Model Measures

The following examples explain how to change Model Measures in several ways:

- Change an existing measure.
- Import an additional standard measure from the model.
- Export an additional custom measure to the model.

Change an existing measure

1. Open the template and select the Configuration worksheet.
2. Edit any Name or Description by highlighting and typing over them.

Note that if you want to change a measure--for example, to prompt the user for Probability instead of Frequency--you only need to change the name and description in Metrics From Model – this is then displayed in the model and the user can choose the appropriate measure to

map.

Import an additional standard measure from the model

1. Open the template and select the Configuration worksheet.
2. From the menus choose:
Tools > Protection > Unprotect Sheet
3. Select cell A5, which is shaded yellow and contains the word End.
4. From the menus choose:
Insert > Rows
5. Type in the Name and Description of the new measure. For example, Error and Error associated with segment.
6. In cell C5, enter the formula =COLUMN('Model Measures'!N3).
7. In cell D5, enter the formula =ROW('Model Measures'!N3)+1.
These formulae will cause the new measure to be displayed in column N of the Model Measures worksheet, which is currently empty.
8. From the menus choose:
Tools > Protection > Protect Sheet
9. Click OK.
10. On the Model Measures worksheet, ensure that cell N3 has Error as a title for the new column.
11. Select all of column N.
12. From the menus choose:
Format > Cells
13. By default, all of the cells have a General number category. Click Percentage to change how the figures are displayed. This helps you check your figures in Excel; in addition, it enables you to utilize the data in other ways, for example, as an output to a graph.
14. Click OK.
15. Save the spreadsheet as an Excel 2003 template, with a unique name and the file extension .xlt. For ease of locating the new template, we recommend you save it in the preconfigured template location on your local or network file system.

Export an additional custom measure to the model

1. Open the template to which you added the Error column in the previous example; select the Configuration worksheet.
2. From the menus choose:
Tools > Protection > Unprotect Sheet
3. Select cell A14, which is shaded yellow and contains the word End.
4. From the menus choose:
Insert > Rows
5. Type in the Name and Description of the new measure. For example, Scaled Error and Scaling applied to error from Excel.
6. In cell C14, enter the formula =COLUMN('Model Measures'!O3).
7. In cell D14, enter the formula =ROW('Model Measures'!O3)+1.
These formulae specify that the column O will supply the new measure to the model.
8. Select the Settings worksheet.
9. In cell A17, enter the description '- Scaled Error.
10. In cell B17, enter the scaling factor of 10.
11. On the Model Measures worksheet, enter the description Scaled Error in cell O3 as a title for the new column.
12. In cell O4, enter the formula =N4*Settings!\$B\$17.
13. Select the corner of cell O4 and drag it down to cell O22 to copy the formula into each cell.
14. From the menus choose:
Tools > Protection > Protect Sheet
15. Click OK.
16. Save the spreadsheet as an Excel 2003 template, with a unique name and the file extension .xlt. For ease of locating the new template, we recommend you save it in the preconfigured template location on your local or network file system.

When you connect to Excel using this template, the Error value is available as a new custom measure.

Visualizing Models

The best way to understand the impact of a model is to visualize it. Using a gains chart, you can obtain valuable day-to-day insight into the business and technical benefit of your model by studying the effect of multiple alternatives in real time. The [Gains Chart](#) section shows the benefit of a model over randomized decision-making and enables the direct comparison of multiple charts when there are alternative models.

- [Gains Chart](#)

Related information

- [Gains Chart](#)
 - [Working with Decision List Viewer](#)
 - [Undo and Redo Actions](#)
 - [Segment Rules](#)
 - [Customizing a Model](#)
 - [Generate New Model](#)
 - [Model Assessment](#)
 - [Assessment in Excel](#)
-

Gains Chart

The gains chart plots the values in the *Gains %* column from the table. Gains are defined as the proportion of hits in each increment relative to the total number of hits in the tree, using the equation:

$$(\text{hits in increment} / \text{total number of hits}) \times 100\%$$

Gains charts effectively illustrate how widely you need to cast the net to capture a given percentage of all of the hits in the tree. The diagonal line plots the expected response for the entire sample if the model is not used. In this case, the response rate would be constant, since one person is just as likely to respond as another. To double your yield, you would need to ask twice as many people. The curved line indicates how much you can improve your response by including only those who rank in the higher percentiles based on gain. For example, including the top 50% might net you more than 70% of the positive responses. The steeper the curve, the higher the gain.

To view a gains chart:

1. Open a stream that contains a Decision List node and launch an interactive session from the node.
2. Click the Gains tab. Depending on which partitions are specified, you may see one or two charts (two charts would display, for example, when both the training and testing partitions are defined for the model measures).

By default, the charts display as segments. You can switch the charts to display as quantiles by selecting Quantiles and then selecting the appropriate quantile method from the drop-down menu.

- [Chart Options](#)

Related information

- [Visualizing Models](#)
-

Chart Options

The Chart Options feature provides options for selecting which models and snapshots are charted, which partitions are plotted, and whether or not segment labels display.

Models to Plot

Current Models. Enables you to select which models to chart. You can select the working model or any created snapshot models.

Partitions to Plot

Partitions for left-hand chart. The drop-down list provides options for displaying all defined partitions or all data.

Partitions for right-hand chart. The drop-down list provides options for displaying all defined partitions, all data, or only the left-hand chart. When Graph only left is selected, only the left chart is displayed.

Display Segment Labels. When enabled, each segment label is displayed on the charts.

Statistical Models

Statistical models use mathematical equations to encode information extracted from the data. In some cases, statistical modeling techniques can provide adequate models very quickly. Even for problems in which more flexible machine-learning techniques (such as neural networks) can

ultimately give better results, you can use some statistical models as baseline predictive models to judge the performance of more advanced techniques.

The following statistical modeling nodes are available.

	Linear regression models predict a continuous target based on linear relationships between the target and one or more predictors.
	Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric range.
	The PCA/Factor node provides powerful data-reduction techniques to reduce the complexity of your data. Principal components analysis (PCA) finds linear combinations of the input fields that do the best job of capturing the variance in the entire set of fields, where the components are orthogonal (perpendicular) to each other. Factor analysis attempts to identify underlying factors that explain the pattern of correlations within a set of observed fields. For both approaches, the goal is to find a small number of derived fields that effectively summarizes the information in the original set of fields.
	Discriminant analysis makes more stringent assumptions than logistic regression but can be a valuable alternative or supplement to a logistic regression analysis when those assumptions are met.
	The Generalized Linear model expands the general linear model so that the dependent variable is linearly related to the factors and covariates through a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers the functionality of a wide number of statistical models, including linear regression, logistic regression, loglinear models for count data, and interval-censored survival models.
	A generalized linear mixed model (GLMM) extends the linear model so that the target can have a non-normal distribution, is linearly related to the factors and covariates via a specified link function, and so that the observations can be correlated. Generalized linear mixed models cover a wide variety of models, from simple linear regression to complex multilevel models for non-normal longitudinal data.
	The Cox regression node enables you to build a survival model for time-to-event data in the presence of censored records. The model produces a survival function that predicts the probability that the event of interest has occurred at a given time (t) for given values of the input variables.

- [Linear Node](#)
- [Linear-AS Node](#)
- [Regression Node](#)
- [Regression Model Nugget](#)
- [Logistic Node](#)
- [Logistic Model Nugget](#)
- [PCA/Factor Node](#)
- [PCA/Factor Model Nugget](#)
- [Discriminant node](#)
- [GenLin Node](#)
- [Generalized Linear Mixed Models](#)
- [GLE Node](#)
- [Cox Node](#)

Related information

- [Screening Fields and Records](#)
- [Anomaly Detection Node](#)
- [Neural Net Node](#)
- [Clustering models](#)
- [Association Rules](#)
- [Time Series Node \(deprecated\)](#)

Linear Node

Linear regression is a common statistical technique for classifying records based on the values of numeric input fields. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values.

Requirements. Only numeric fields can be used in a linear regression model. You must have exactly one target field (with the role set to Target) and one or more predictors (with the role set to Input). Fields with a role of Both or None are ignored, as are non-numeric fields. (If necessary, non-numeric fields can be recoded using a Derive node.)

Strengths. Linear regression models are relatively simple and give an easily interpreted mathematical formula for generating predictions. Because linear regression is a long-established statistical procedure, the properties of these models are well understood. Linear models are also

typically very fast to train. The Linear node provides methods for automatic field selection in order to eliminate nonsignificant input fields from the equation.

Note: In cases where the target field is categorical rather than a continuous range, such as yes/no or churn/don't churn, logistic regression can be used as an alternative. Logistic regression also provides support for non-numeric inputs, removing the need to recode these fields. See the topic [Logistic Node](#) for more information.

- [Linear models](#)
-

Linear models

Linear models predict a continuous target based on linear relationships between the target and one or more predictors.

Linear models are relatively simple and give an easily interpreted mathematical formula for scoring. The properties of these models are well understood and can typically be built very quickly compared to other model types (such as neural networks or decision trees) on the same dataset.

Example. An insurance company with limited resources to investigate homeowners' insurance claims wants to build a model for estimating claims costs. By deploying this model to service centers, representatives can enter claim information while on the phone with a customer and immediately obtain the "expected" cost of the claim based on past data.

Field requirements. There must be a Target and at least one Input. By default, fields with predefined roles of Both or None are not used. The target must be continuous (scale). There are no measurement level restrictions on predictors (inputs); categorical (flag, nominal, and ordinal) fields are used as factors in the model and continuous fields are used as covariates.

- [Objectives \(linear models\)](#)
 - [Basics \(linear models\)](#)
 - [Model selection \(linear models\)](#)
 - [Ensembles \(linear models\)](#)
 - [Advanced \(linear models\)](#)
 - [Model Options \(linear models\)](#)
 - [Model Summary \(linear models\)](#)
 - [Automatic Data Preparation \(linear models\)](#)
 - [Predictor Importance \(linear models\)](#)
 - [Predicted By Observed \(linear models\)](#)
 - [Residuals \(linear models\)](#)
 - [Outliers \(linear models\)](#)
 - [Effects \(linear models\)](#)
 - [Coefficients \(linear models\)](#)
 - [Estimated means \(linear models\)](#)
 - [Model Building Summary \(linear models\)](#)
 - [Settings \(linear models\)](#)
-

Objectives (linear models)

What do you want to do?

- Build a new model. Build a completely new model. This is the usual operation of the node.
- Continue training an existing model. Training continues with the last model successfully produced by the node. This makes it possible to update or refresh an existing model without having to access the original data and may result in significantly faster performance since only the new or updated records are fed into the stream. Details on the previous model are stored with the modeling node, making it possible to use this option even if the previous model nugget is no longer available in the stream or Models palette.

Note: When this option is enabled, all other controls on the Fields and Build Options tabs are disabled.

What is your main objective? Select the appropriate objective.

- Create a standard model. The method builds a single model to predict the target using the predictors. Generally speaking, standard models are easier to interpret and can be faster to score than boosted, bagged, or large dataset ensembles.
Note: Continue training existing model is only supported with Build a single tree split models, and you must be connected to Analytic Server.
- Enhance model accuracy (boosting). The method builds an ensemble model using boosting, which generates a sequence of models to obtain more accurate predictions. Ensembles can take longer to build and to score than a standard model.
Boosting produces a succession of "component models", each of which is built on the entire dataset. Prior to building each successive component model, the records are weighted based on the previous component model's residuals. Cases with large residuals are given

relatively higher analysis weights so that the next component model will focus on predicting these records well. Together these component models form an ensemble model. The ensemble model scores new records using a combining rule; the available rules depend upon the measurement level of the target.

- Enhance model stability (bagging). The method builds an ensemble model using bagging (bootstrap aggregating), which generates multiple models to obtain more reliable predictions. Ensembles can take longer to build and to score than a standard model. Bootstrap aggregation (bagging) produces replicates of the training dataset by sampling with replacement from the original dataset. This creates bootstrap samples of equal size to the original dataset. Then a "component model" is built on each replicate. Together these component models form an ensemble model. The ensemble model scores new records using a combining rule; the available rules depend upon the measurement level of the target.
- Create a model for very large datasets. The method builds an ensemble model by splitting the dataset into separate data blocks. Choose this option if your dataset is too large to build any of the models above, or for incremental model building. This option can take less time to build, but can take longer to score than a standard model.

See [Ensembles \(linear models\)](#) for settings related to boosting, bagging, and very large datasets.

Basics (linear models)

Automatically prepare data. This option allows the procedure to internally transform the target and predictors in order to maximize the predictive power of the model; any transformations are saved with the model and applied to new data for scoring. The original versions of transformed fields are excluded from the model. By default, the following automatic data preparation are performed.

- Date and Time handling. Each date predictor is transformed into new a continuous predictor containing the elapsed time since a reference date (1970-01-01). Each time predictor is transformed into a new continuous predictor containing the time elapsed since a reference time (00:00:00).
- Adjust measurement level. Continuous predictors with less than 5 distinct values are recast as ordinal predictors. Ordinal predictors with greater than 10 distinct values are recast as continuous predictors.
- Outlier handling. Values of continuous predictors that lie beyond a cutoff value (3 standard deviations from the mean) are set to the cutoff value.
- Missing value handling. Missing values of nominal predictors are replaced with the mode of the training partition. Missing values of ordinal predictors are replaced with the median of the training partition. Missing values of continuous predictors are replaced with the mean of the training partition.
- Supervised merging. This makes a more parsimonious model by reducing the number of fields to be processed in association with the target. Similar categories are identified based upon the relationship between the input and the target. Categories that are not significantly different (that is, having a p-value greater than 0.1) are merged. If all categories are merged into one, the original and derived versions of the field are excluded from the model because they have no value as a predictor.

Confidence level. This is the level of confidence used to compute interval estimates of the model coefficients in the [Coefficients](#) view. Specify a value greater than 0 and less than 100. The default is 95.

Model selection (linear models)

Model selection method. Choose one of the model selection methods (details below) or Include all predictors, which simply enters all available predictors as main effects model terms. By default, Forward stepwise is used.

Forward Stepwise Selection. This starts with no effects in the model and adds and removes effects one step at a time until no more can be added or removed according to the stepwise criteria.

- Criteria for entry/removal. This is the statistic used to determine whether an effect should be added to or removed from the model. Information Criterion (AICC) is based on the likelihood of the training set given the model, and is adjusted to penalize overly complex models. F Statistics is based on a statistical test of the improvement in model error. Adjusted R-squared is based on the fit of the training set, and is adjusted to penalize overly complex models. Overfit Prevention Criterion (ASE) is based on the fit (average squared error, or ASE) of the overfit prevention set. The overfit prevention set is a random subsample of approximately 30% of the original dataset that is not used to train the model.

If any criterion other than F Statistics is chosen, then at each step the effect that corresponds to the greatest positive increase in the criterion is added to the model. Any effects in the model that correspond to a decrease in the criterion are removed.

If F Statistics is chosen as the criterion, then at each step the effect that has the smallest p-value less than the specified threshold, Include effects with p-values less than, is added to the model. The default is 0.05. Any effects in the model with a p-value greater than the specified threshold, Remove effects with p-values greater than, are removed. The default is 0.10.

- Customize maximum number of effects in the final model. By default, all available effects can be entered into the model. Alternatively, if the stepwise algorithm ends a step with the specified maximum number of effects, the algorithm stops with the current set of effects.

- Customize maximum number of steps. The stepwise algorithm stops after a certain number of steps. By default, this is 3 times the number of available effects. Alternatively, specify a positive integer maximum number of steps.

Best Subsets Selection. This checks "all possible" models, or at least a larger subset of the possible models than forward stepwise, to choose the best according to the best subsets criterion. Information Criterion (AICC) is based on the likelihood of the training set given the model, and is adjusted to penalize overly complex models. Adjusted R-squared is based on the fit of the training set, and is adjusted to penalize overly complex models. Overfit Prevention Criterion (ASE) is based on the fit (average squared error, or ASE) of the overfit prevention set. The overfit prevention set is a random subsample of approximately 30% of the original dataset that is not used to train the model.

The model with the greatest value of the criterion is chosen as the best model.

Note: Best subsets selection is more computationally intensive than forward stepwise selection. When best subsets is performed in conjunction with boosting, bagging, or very large datasets, it can take considerably longer to build than a standard model built using forward stepwise selection.

Ensembles (linear models)

These settings determine the behavior of ensembling that occurs when boosting, bagging, or very large datasets are requested in Objectives. Options that do not apply to the selected objective are ignored.

Bagging and Very Large Datasets. When scoring an ensemble, this is the rule used to combine the predicted values from the base models to compute the ensemble score value.

- Default combining rule for continuous targets. Ensemble predicted values for continuous targets can be combined using the mean or median of the predicted values from the base models.

Note that when the objective is to enhance model accuracy, the combining rule selections are ignored. Boosting always uses a weighted majority vote to score categorical targets and a weighted median to score continuous targets.

Boosting and Bagging. Specify the number of base models to build when the objective is to enhance model accuracy or stability; for bagging, this is the number of bootstrap samples. It should be a positive integer.

Advanced (linear models)

Replicate results. Setting a random seed allows you to replicate analyses. The random number generator is used to choose which records are in the overfit prevention set. Specify an integer or click Generate, which will create a pseudo-random integer between 1 and 2147483647, inclusive. The default is 54752075.

Model Options (linear models)

Model Name. You can generate the model name automatically based on the target fields or specify a custom name. The automatically generated name is the target field name.

Note that the predicted value is always computed when the model is scored. The name of the new field is the name of the target field, prefixed with \$L-. For example, for a target field named *sales*, the new field would be named *\$L-sales*.

Model Summary (linear models)

The Model Summary view is a snapshot, at-a-glance summary of the model and its fit.

Table

The table identifies some high-level model settings, including:

- The name of the target specified on the [Fields](#) tab,
- Whether automatic data preparation was performed as specified on the [Basics](#) settings,
- The model selection method and selection criterion specified on the [Model Selection](#) settings. The value of the selection criterion for the final model is also displayed, and is presented in smaller is better format.

Chart

The chart displays the accuracy of the final model, which is presented in larger is better format. The value is $100 \times$ the adjusted R^2 for the final model.

Automatic Data Preparation (linear models)

This view shows information about which fields were excluded and how transformed fields were derived in the automatic data preparation (ADP) step. For each field that was transformed or excluded, the table lists the field name, its role in the analysis, and the action taken by the ADP step. Fields are sorted by ascending alphabetical order of field names. The possible actions taken for each field include:

- Derive duration: months computes the elapsed time in months from the values in a field containing dates to the current system date.
 - Derive duration: hours computes the elapsed time in hours from the values in a field containing times to the current system time.
 - Change measurement level from continuous to ordinal recasts continuous fields with less than 5 unique values as ordinal fields.
 - Change measurement level from ordinal to continuous recasts ordinal fields with more than 10 unique values as continuous fields.
 - Trim outliers sets values of continuous predictors that lie beyond a cutoff value (3 standard deviations from the mean) to the cutoff value.
 - Replace missing values replaces missing values of nominal fields with the mode, ordinal fields with the median, and continuous fields with the mean.
 - Merge categories to maximize association with target identifies "similar" predictor categories based upon the relationship between the input and the target. Categories that are not significantly different (that is, having a p -value greater than 0.05) are merged.
 - Exclude constant predictor / after outlier handling / after merging of categories removes predictors that have a single value, possibly after other ADP actions have been taken.
-

Predictor Importance (linear models)

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

Predicted By Observed (linear models)

This displays a binned scatterplot of the predicted values on the vertical axis by the observed values on the horizontal axis. Ideally, the points should lie on a 45-degree line; this view can tell you whether any records are predicted particularly badly by the model.

Residuals (linear models)

This displays a diagnostic chart of model residuals.

Chart styles. There are different display styles, which are accessible from the Style drop-down.

- Histogram. This is a binned histogram of the studentized residuals with an overlay of the normal distribution. Linear models assume that the residuals have a normal distribution, so the histogram should ideally closely approximate the smooth line.
 - P-P Plot. This is a binned probability-probability plot comparing the studentized residuals to a normal distribution. If the slope of the plotted points is less steep than the normal line, the residuals show greater variability than a normal distribution; if the slope is steeper, the residuals show less variability than a normal distribution. If the plotted points have an S-shaped curve, then the distribution of residuals is skewed.
-

Outliers (linear models)

This table lists records that exert undue influence upon the model, and displays the record ID (if specified on the Fields tab), target value, and Cook's distance. Cook's distance is a measure of how much the residuals of all records would change if a particular record were excluded from the calculation of the model coefficients. A large Cook's distance indicates that excluding a record from changes the coefficients substantially, and should therefore be considered influential.

Influential records should be examined carefully to determine whether you can give them less weight in estimating the model, or truncate the outlying values to some acceptable threshold, or remove the influential records completely.

Effects (linear models)

This view displays the size of each effect in the model.

Styles. There are different display styles, which are accessible from the Style dropdown list.

- Diagram. This is a chart in which effects are sorted from top to bottom by decreasing predictor importance. Connecting lines in the diagram are weighted based on effect significance, with greater line width corresponding to more significant effects (smaller p -values). Hovering over a connecting line reveals a tooltip that shows the p -value and importance of the effect. This is the default.
- Table. This is an ANOVA table for the overall model and the individual model effects. The individual effects are sorted from top to bottom by decreasing predictor importance. Note that by default, the table is collapsed to only show the results for the overall model. To see the results for the individual model effects, click the Corrected Model cell in the table.

Predictor importance. There is a Predictor Importance slider that controls which predictors are shown in the view. This does not change the model, but simply allows you to focus on the most important predictors. By default, the top 10 effects are displayed.

Significance. There is a Significance slider that further controls which effects are shown in the view, beyond those shown based on predictor importance. Effects with significance values greater than the slider value are hidden. This does not change the model, but simply allows you to focus on the most important effects. By default the value is 1.00, so that no effects are filtered based on significance.

Coefficients (linear models)

This view displays the value of each coefficient in the model. Note that factors (categorical predictors) are indicator-coded within the model, so that **effects** containing factors will generally have multiple associated **coefficients**; one for each category except the category corresponding to the redundant (reference) parameter.

Styles. There are different display styles, which are accessible from the Style dropdown list.

- Diagram. This is a chart which displays the intercept first, and then sorts effects from top to bottom by decreasing predictor importance. Within effects containing factors, coefficients are sorted by ascending order of data values. Connecting lines in the diagram are colored based on the sign of the coefficient (see the diagram key) and weighted based on coefficient significance, with greater line width corresponding to more significant coefficients (smaller p -values). Hovering over a connecting line reveals a tooltip that shows the value of the coefficient, its p -value, and the importance of the effect the parameter is associated with. This is the default style.
- Table. This shows the values, significance tests, and confidence intervals for the individual model coefficients. After the intercept, the effects are sorted from top to bottom by decreasing predictor importance. Within effects containing factors, coefficients are sorted by ascending order of data values. Note that by default the table is collapsed to only show the coefficient, significance, and importance of each model parameter. To see the standard error, t statistic, and confidence interval, click the Coefficient cell in the table. Hovering over the name of a model parameter in the table reveals a tooltip that shows the name of the parameter, the effect the parameter is associated with, and (for categorical predictors), the value labels associated with the model parameter. This can be particularly useful to see the new categories created when automatic data preparation merges similar categories of a categorical predictor.

Predictor importance. There is a Predictor Importance slider that controls which predictors are shown in the view. This does not change the model, but simply allows you to focus on the most important predictors. By default, the top 10 effects are displayed.

Significance. There is a Significance slider that further controls which coefficients are shown in the view, beyond those shown based on predictor importance. Coefficients with significance values greater than the slider value are hidden. This does not change the model, but simply allows you to focus on the most important coefficients. By default the value is 1.00, so that no coefficients are filtered based on significance.

Estimated means (linear models)

These are charts displayed for significant predictors. The chart displays the model-estimated value of the target on the vertical axis for each value of the predictor on the horizontal axis, holding all other predictors constant. It provides a useful visualization of the effects of each predictor's coefficients on the target.

Note: If no predictors are significant, no estimated means are produced.

Model Building Summary (linear models)

When a model selection algorithm other than None is chosen on the Model Selection settings, this provides some details of the model building process.

Forward stepwise. When forward stepwise is the selection algorithm, the table displays the last 10 steps in the stepwise algorithm. For each step, the value of the selection criterion and the effects in the model at that step are shown. This gives you a sense of how much each step contributes to the model. Each column allows you to sort the rows so that you can more easily see which effects are in the model at a given step.

Best subsets. When best subsets is the selection algorithm, the table displays the top 10 models. For each model, the value of the selection criterion and the effects in the model are shown. This gives you a sense of the stability of the top models; if they tend to have many similar effects with a few differences, then you can be fairly confident in the "top" model; if they tend to have very different effects, then some of the effects may be too similar and should be combined (or one removed). Each column allows you to sort the rows so that you can more easily see which effects are in the model at a given step.

Settings (linear models)

Note that the predicted value is always computed when the model is scored. The name of the new field is the name of the target field, prefixed with `$L-`. For example, for a target field named `sales`, the new field would be named `$L-sales`.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score by converting to native SQL If selected, generates native SQL to score the model within the database.
Note: Although this option can provide quicker results, the size and complexity of the native SQL increases as the complexity of the model increases.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Linear-AS Node

IBM® SPSS® Modeler has two different versions of the Linear node:

- Linear is the traditional node that runs on the IBM SPSS Modeler Server.
- Linear-AS can run when connected to IBM SPSS Analytic Server.

Linear regression is a common statistical technique for classifying records based on the values of numeric input fields. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values.

Requirements. Only numeric fields and categorical predictors can be used in a linear regression model. You must have exactly one target field (with the role set to Target) and one or more predictors (with the role set to Input). Fields with a role of Both or None are ignored, as are non-numeric fields. (If necessary, non-numeric fields can be recoded using a Derive node.)

Strengths. Linear regression models are relatively simple and give an easily interpreted mathematical formula for generating predictions. Because linear regression is a long-established statistical procedure, the properties of these models are well understood. Linear models are also typically very fast to train. The Linear node provides methods for automatic field selection in order to eliminate nonsignificant input fields from the equation.

Note: In cases where the target field is categorical rather than a continuous range, such as yes/no or churn/don't churn, logistic regression can be used as an alternative. Logistic regression also provides support for non-numeric inputs, removing the need to recode these fields. See the topic [Logistic Node](#) for more information.

- [Linear-AS models](#)

Related information

- [Linear-AS models](#)
- [Linear Node](#)

Linear-AS models

Linear models predict a continuous target based on linear relationships between the target and one or more predictors.

Linear models are relatively simple and give an easily interpreted mathematical formula for scoring. The properties of these models are well understood and can typically be built very quickly compared to other model types (such as neural networks or decision trees) on the same dataset.

Example. An insurance company with limited resources to investigate homeowners' insurance claims wants to build a model for estimating claims costs. By deploying this model to service centers, representatives can enter claim information while on the phone with a customer and immediately obtain the "expected" cost of the claim based on past data.

Field requirements. There must be a Target and at least one Input. By default, fields with predefined roles of Both or None are not used. The target must be continuous (scale). There are no measurement level restrictions on predictors (inputs); categorical (flag, nominal, and ordinal) fields are used as factors in the model and continuous fields are used as covariates.

- [Basics \(linear-AS models\)](#)
 - [Model Selection \(linear-AS models\)](#)
 - [Model Options \(linear-AS models\)](#)
 - [Interactive Output \(linear-AS models\)](#)
 - [Settings \(linear-AS models\)](#)
-

Basics (linear-AS models)

Include intercept. This option includes an offset on the y axis when the x axis is 0. The intercept is usually included in the model. If you can assume the data pass through the origin, you can exclude the intercept.

Consider two way interaction. This option tells the model to compare each possible pair of inputs to see if the trend of one affects the other. If it does, then those inputs are more likely to be included in the design matrix.

Confidence interval for coefficient estimates (%). This is the interval of confidence used to compute estimates of the model coefficients in the [Coefficients](#) view. Specify a value greater than 0 and less than 100. The default is 95.

Sorting order for categorical predictors. These controls determine the order of the categories for the factors (categorical inputs) for purposes of determining the "last" category. The sort order setting is ignored if the input is not categorical or if a custom reference category is specified.

Model Selection (linear-AS models)

Model selection method. Choose one of the model selection methods (details below) or Include all predictors, which simply enters all available predictors as main effects model terms. By default, Forward stepwise is used.

Forward Stepwise Selection. This starts with no effects in the model and adds and removes effects one step at a time until no more can be added or removed according to the stepwise criteria.

- Criteria for entry/removal. This is the statistic used to determine whether an effect should be added to or removed from the model. Information Criterion (AICC) is based on the likelihood of the training set given the model, and is adjusted to penalize overly complex models. F Statistics is based on a statistical test of the improvement in model error. Adjusted R-squared is based on the fit of the training set, and is adjusted to penalize overly complex models. Overfit Prevention Criterion (ASE) is based on the fit (average squared error, or ASE) of the overfit prevention set. The overfit prevention set is a random subsample of approximately 30% of the original dataset that is not used to train the model.

If any criterion other than F Statistics is chosen, then at each step the effect that corresponds to the greatest positive increase in the criterion is added to the model. Any effects in the model that correspond to a decrease in the criterion are removed.

If F Statistics is chosen as the criterion, then at each step the effect that has the smallest *p*-value less than the specified threshold, Include effects with *p*-values less than, is added to the model. The default is 0.05. Any effects in the model with a *p*-value greater than the specified threshold, Remove effects with *p*-values greater than, are removed. The default is 0.10.

- Customize maximum number of effects in the final model. By default, all available effects can be entered into the model. Alternatively, if the stepwise algorithm ends a step with the specified maximum number of effects, the algorithm stops with the current set of effects.
- Customize maximum number of steps. The stepwise algorithm stops after a certain number of steps. By default, this is 3 times the number of available effects. Alternatively, specify a positive integer maximum number of steps.

Best Subsets Selection. This checks "all possible" models, or at least a larger subset of the possible models than forward stepwise, to choose the best according to the best subsets criterion. Information Criterion (AICC) is based on the likelihood of the training set given the model, and is adjusted to penalize overly complex models. Adjusted R-squared is based on the fit of the training set, and is adjusted to penalize overly complex models. Overfit Prevention Criterion (ASE) is based on the fit (average squared error, or ASE) of the overfit prevention set. The overfit prevention set is a random subsample of approximately 30% of the original dataset that is not used to train the model.

The model with the greatest value of the criterion is chosen as the best model.

Note: Best subsets selection is more computationally intensive than forward stepwise selection. When best subsets is performed in conjunction with boosting, bagging, or very large datasets, it can take considerably longer to build than a standard model built using forward stepwise selection.

Model Options (linear-AS models)

Model Name. You can generate the model name automatically based on the target fields or specify a custom name. The automatically generated name is the target field name.

Note that the predicted value is always computed when the model is scored. The name of the new field is the name of the target field, prefixed with $\$L$ - . For example, for a target field named *sales*, the new field would be named $\$L\text{-}sales$.

Interactive Output (linear-AS models)

After running a Linear-AS model, the following output is available.

Model Information

The Model Information view provides key information about the model. The table identifies some high-level model settings, such as:

- The name of the target specified on the [Fields](#) tab
- The regression weight field
- The model building method specified on the [Model Selection](#) settings
- The number of predictors input
- The number of predictors in the final model
- Akaike Information Criterion Corrected (AICC). AICC is a measure for selecting and comparing mixed models based on the -2 (Restricted) log likelihood. Smaller values indicate better models. The AICC "corrects" the AIC for small sample sizes. As the sample size increases, the AICC converges to the AIC.
- R Square. This is the goodness-of-fit measure of a linear model, sometimes called the coefficient of determination. It is the proportion of variation in the dependent variable explained by the regression model. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well.
- Adjusted R Square

Records Summary

The Records Summary view provides information about the number and percentage of records (cases) included and excluded from the model.

Predictor Importance

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

Predicted by Observed

This displays a binned scatterplot of the predicted values on the vertical axis by the observed values on the horizontal axis. Ideally, the points should lie on a 45-degree line; this view can tell you whether any records are predicted particularly badly by the model.

Settings (linear-AS models)

Note that the predicted value is always computed when the model is scored. The name of the new field is the name of the target field, prefixed with $\$L$ - . For example, for a target field named *sales*, the new field would be named $\$L\text{-}sales$.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process. If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
 - Score outside of the Database. If selected, this option fetches your data back from the database and scores it in SPSS Modeler.
-

Logistic Node

Logistic regression, also known as **nominal regression**, is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric one. Both binomial models (for targets with two discrete categories) and multinomial models (for targets with more than two categories) are supported.

Logistic regression works by building a set of equations that relate the input field values to the probabilities associated with each of the output field categories. Once the model is generated, it can be used to estimate probabilities for new data. For each record, a probability of membership is computed for each possible output category. The target category with the highest probability is assigned as the predicted output value for that record.

Binomial example. A telecommunications provider is concerned about the number of customers it is losing to competitors. Using service usage data, you can create a binomial model to predict which customers are liable to transfer to another provider and customize offers so as to retain as many customers as possible. A binomial model is used because the target has two distinct categories (likely to transfer or not).

Note: For binomial models only, string fields are limited to eight characters. If necessary, longer strings can be recoded using a Reclassify node or by using the Anonymize node.

Multinomial example. A telecommunications provider has segmented its customer base by service usage patterns, categorizing the customers into four groups. Using demographic data to predict group membership, you can create a multinomial model to classify prospective customers into groups and then customize offers for individual customers.

Requirements. One or more input fields and exactly one categorical target field with two or more categories. For a binomial model the target must have a measurement level of *Flag*. For a multinomial model the target can have a measurement level of *Flag*, or of *Nominal* with two or more categories. Fields set to *Both* or *None* are ignored. Fields used in the model must have their types fully instantiated.

Strengths. Logistic regression models are often quite accurate. They can handle symbolic and numeric input fields. They can give predicted probabilities for all target categories so that a second-best guess can easily be identified. Logistic models are most effective when group membership is a truly categorical field; if group membership is based on values of a continuous range field (for example, high IQ versus low IQ), you should consider using linear regression to take advantage of the richer information offered by the full range of values. Logistic models can also perform automatic field selection, although other approaches such as tree models or Feature Selection might do this more quickly on large datasets. Finally, since logistic models are well understood by many analysts and data miners, they may be used by some as a baseline against which other modeling techniques can be compared.

When processing large datasets, you can improve performance noticeably by disabling the likelihood-ratio test, an advanced output option. See the topic [Logistic Regression Advanced Output](#) for more information.

Important: If temporary disk space is low, Binomial Logistic Regression can fail to build, and will display an error. When building from a large data set (10GB or more), the same amount of free disk space is needed. You can use the environment variable SPSSTMPDIR to set the location of the temporary directory.

- [Logistic Node Model Options](#)
- [Adding Terms to a Logistic Regression Model](#)
- [Logistic Node Expert Options](#)
- [Logistic Regression Convergence Options](#)
- [Logistic Regression Advanced Output](#)
- [Logistic Regression Stepping Options](#)

Related information

- [Overview of modeling nodes](#)
 - [Modeling Node Fields Options](#)
 - [Logistic Node Model Options](#)
 - [Adding Terms to a Logistic Regression Model](#)
 - [Logistic Node Expert Options](#)
 - [Logistic Regression Convergence Options](#)
 - [Logistic Regression Advanced Output](#)
 - [Logistic Regression Stepping Options](#)
-

Logistic Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Create split models. Builds a separate model for each possible value of input fields that are specified as split fields. See [Building Split Models](#) for more information.

Procedure. Specifies whether a binomial or multinomial model is created. The options available in the dialog box vary depending on which type of modeling procedure is selected.

- **Binomial.** Used when the target field is a flag or nominal field with two discrete values (dichotomous), such as yes/no, on/off, male/female.
- **Multinomial.** Used when the target field is a nominal field with more than two values. You can specify Main effects, Full factorial, or Custom.

Include constant in equation. This option determines whether the resulting equations will include a constant term. In most situations, you should leave this option selected.

Binomial Models

For binomial models, the following methods and options are available:

Method. Specify the method to be used in building the logistic regression model.

- **Enter.** This is the default method, which enters all of the terms into the equation directly. No field selection is performed in building the model.
- **Forwards Stepwise.** The Forwards Stepwise method of field selection builds the equation in steps, as the name implies. The initial model is the simplest model possible, with no model terms (except the constant) in the equation. At each step, terms that have not yet been added to the model are evaluated, and if the best of those terms adds significantly to the predictive power of the model, it is added. In addition, terms that are currently in the model are reevaluated to determine if any of them can be removed without significantly detracting from the model. If so, they are removed. The process repeats, and other terms are added and/or removed. When no more terms can be added to improve the model, and no more terms can be removed without detracting from the model, the final model is generated.
- **Backwards Stepwise.** The Backwards Stepwise method is essentially the opposite of the Forwards Stepwise method. With this method, the initial model contains all of the terms as predictors. At each step, terms in the model are evaluated, and any terms that can be removed without significantly detracting from the model are removed. In addition, previously removed terms are reevaluated to determine if the best of those terms adds significantly to the predictive power of the model. If so, it is added back into the model. When no more terms can be removed without significantly detracting from the model, and no more terms can be added to improve the model, the final model is generated.

Categorical inputs. Lists the fields that are identified as categorical, that is, those with a measurement level of flag, nominal, or ordinal. You can specify the contrast and base category for each categorical field.

- **Field Name.** This column contains the field names of the categorical inputs. To add continuous or numerical inputs into this column, click the Add Fields icon to the right of the list and select the required inputs.
- **Contrast.** The interpretation of the regression coefficients for a categorical field depends on the contrasts that are used. The contrast determines how hypothesis tests are set up to compare the estimated means. For example, if you know that a categorical field has implicit order, such as a pattern or grouping, you can use the contrast to model that order. The available contrasts are:
Indicator. Contrasts indicate the presence or absence of category membership. This is the default method.

Simple. Each category of the predictor field, except the reference category, is compared to the reference category.

Difference. Each category of the predictor field, except the first category, is compared to the average effect of previous categories. Also known as reverse Helmert contrasts.

Helmert. Each category of the predictor field, except the last category, is compared to the average effect of subsequent categories.

Repeated. Each category of the predictor field, except the first category, is compared to the category that precedes it.

Polynomial. Orthogonal polynomial contrasts. Categories are assumed to be equally spaced. Polynomial contrasts are available for numeric fields only.

Deviation. Each category of the predictor field, except the reference category, is compared to the overall effect.

- **Base Category.** Specifies how the reference category is determined for the selected contrast type. Select First to use the first category for the input field—sorted alphabetically—or select Last to use the last category. The default base category applies to variables that are listed in the Categorical inputs area.

Note: This field is unavailable if the contrast setting is Difference, Helmert, Repeated, or Polynomial.

The estimate of each field's effect on the overall response is computed as an increase or decrease in the likelihood of each of the other categories relative to the reference category. This can help you identify the fields and values that are more likely to give a specific response.

The base category is shown in the output as 0.0. This is because comparing it to itself produces an empty result. All other categories are shown as equations relevant to the base category. See the topic [Logistic Nugget Model Details](#) for more information.

Multinomial Models

For multinomial models the following methods and options are available:

Method. Specify the method to be used in building the logistic regression model.

- **Enter.** This is the default method, which enters all of the terms into the equation directly. No field selection is performed in building the model.
- **Stepwise.** The Stepwise method of field selection builds the equation in steps, as the name implies. The initial model is the simplest model possible, with no model terms (except the constant) in the equation. At each step, terms that have not yet been added to the model are evaluated, and if the best of those terms adds significantly to the predictive power of the model, it is added. In addition, terms that are currently in the model are reevaluated to determine if any of them can be removed without significantly detracting from the model. If so, they are removed. The process repeats, and other terms are added and/or removed. When no more terms can be added to improve the model, and no more terms can be removed without detracting from the model, the final model is generated.
- **Forwards.** The Forwards method of field selection is similar to the Stepwise method in that the model is built in steps. However, with this method, the initial model is the simplest model, and only the constant and terms can be added to the model. At each step, terms not yet in the model are tested based on how much they would improve the model, and the best of those terms is added to the model. When no more terms can be added, or the best candidate term does not produce a large-enough improvement in the model, the final model is generated.
- **Backwards.** The Backwards method is essentially the opposite of the Forwards method. With this method, the initial model contains all of the terms as predictors, and terms can only be removed from the model. Model terms that contribute little to the model are removed one by one until no more terms can be removed without significantly worsening the model, yielding the final model.
- **Backwards Stepwise.** The Backwards Stepwise method is essentially the opposite of the Stepwise method. With this method, the initial model contains all of the terms as predictors. At each step, terms in the model are evaluated, and any terms that can be removed without significantly detracting from the model are removed. In addition, previously removed terms are reevaluated to determine if the best of those terms adds significantly to the predictive power of the model. If so, it is added back into the model. When no more terms can be removed without significantly detracting from the model, and no more terms can be added to improve the model, the final model is generated.

Note: The automatic methods, including Stepwise, Forwards, and Backwards, are highly adaptable learning methods and have a strong tendency to overfit the training data. When using these methods, it is especially important to verify the validity of the resulting model either with new data or a hold-out test sample created using the Partition node.

Base category for target. Specifies how the reference category is determined. This is used as the baseline against which the regression equations for all other categories in the target are estimated. Select First to use the first category for the current target field—sorted alphabetically—or select Last to use the last category. Alternatively, you can select Specify to choose a specific category and select the desired value from the list. Available values can be defined for each field in a Type node.

Often you would specify the category in which you are least interested to be the base category, for example, a loss-leader product. The other categories are then related to this base category in a relative fashion to identify what makes them more likely to be in their own category. This can help you identify the fields and values that are more likely to give a specific response.

The base category is shown in the output as 0.0. This is because comparing it to itself produces an empty result. All other categories are shown as equations relevant to the base category. See the topic [Logistic Nugget Model Details](#) for more information.

Model type. There are three options for defining the terms in the model. **Main Effects** models include only the input fields individually and do not test interactions (multiplicative effects) between input fields. **Full Factorial** models include all interactions as well as the input field main effects. Full factorial models are better able to capture complex relationships but are also much more difficult to interpret and are more likely to suffer from overfitting. Because of the potentially large number of possible combinations, automatic field selection methods (methods other than Enter) are disabled for full factorial models. **Custom** models include only the terms (main effects and interactions) that you specify. When selecting this option, use the Model Terms list to add or remove terms in the model.

Model Terms. When building a Custom model, you will need to explicitly specify the terms in the model. The list shows the current set of terms for the model. The buttons on the right side of the Model Terms list enable you to add and remove model terms.

- To add terms to the model, click the *Add new model terms* button. See the topic [Adding Terms to a Logistic Regression Model](#) for more information.
- To delete terms, select the desired terms and click the *Delete selected model terms* button.

Related information

- [Logistic Node](#)
- [Adding Terms to a Logistic Regression Model](#)
- [Logistic Node Expert Options](#)
- [Logistic Regression Convergence Options](#)
- [Logistic Regression Advanced Output](#)
- [Logistic Regression Stepping Options](#)

Adding Terms to a Logistic Regression Model

When requesting a custom logistic regression model, you can add terms to the model by clicking the *Add new model terms* button on the Logistic Regression Model tab. The New Terms dialog box opens in which you can specify terms.

Type of term to add. There are several ways to add terms to the model, based on the selection of input fields in the Available fields list.

- **Single interaction.** Inserts the term representing the interaction of all selected fields.
- **Main effects.** Inserts one main effect term (the field itself) for each selected input field.
- **All 2-way interactions.** Inserts a 2-way interaction term (the product of the input fields) for each possible pair of selected input fields. For example, if you have selected input fields A, B, and C in the Available fields list, this method will insert the terms $A * B$, $A * C$, and $B * C$.
- **All 3-way interactions.** Inserts a 3-way interaction term (the product of the input fields) for each possible combination of selected input fields, taken three at a time. For example, if you have selected input fields A, B, C, and D in the Available fields list, this method will insert the terms $A * B * C$, $A * B * D$, $A * C * D$, and $B * C * D$.
- **All 4-way interactions.** Inserts a 4-way interaction term (the product of the input fields) for each possible combination of selected input fields, taken four at a time. For example, if you have selected input fields A, B, C, D, and E in the Available fields list, this method will insert the terms $A * B * C * D$, $A * B * C * E$, $A * B * D * E$, $A * C * D * E$, and $B * C * D * E$.

Available fields. Lists the available input fields to be used in constructing model terms.

Preview. Shows the terms that will be added to the model if you click Insert, based on the selected fields and term type.

Insert. Inserts terms in the model (based on the current selection of fields and term type) and closes the dialog box.

Related information

- [Logistic Node](#)
- [Logistic Node Model Options](#)
- [Logistic Node Expert Options](#)
- [Logistic Regression Convergence Options](#)
- [Logistic Regression Advanced Output](#)
- [Logistic Regression Stepping Options](#)

Logistic Node Expert Options

If you have detailed knowledge of logistic regression, expert options enable you to fine-tune the training process. To access expert options, set Mode to Expert on the Expert tab.

Scale (Multinomial models only). You can specify a dispersion scaling value that will be used to correct the estimate of the parameter covariance matrix. Pearson estimates the scaling value by using the Pearson chi-square statistic. Deviance estimates the scaling value by using the deviance function (likelihood-ratio chi-square) statistic. You can also specify your own user-defined scaling value. It must be a positive numeric value.

Append all probabilities. If this option is selected, probabilities for each category of the output field will be added to each record processed by the node. If this option is not selected, the probability of only the predicted category is added.

For example, a table containing the results of a multinomial model with three categories will include five new columns. One column will list the probability of the outcome being correctly predicted, the next column will show the probability that this prediction is a hit or miss, and a further three columns will show the probability that each category's prediction is a miss or hit. See the topic [Logistic Model Nugget](#) for more information.

Note: This option is always selected for binomial models.

Singularity tolerance. Specify the tolerance used in checking for singularities.

Convergence. These options enable you to control the parameters for model convergence. When you execute the model, the convergence settings control how many times the different parameters are repeatedly run through to see how well they fit. The more often the parameters are tried, the closer the results will be (that is, the results will converge). See the topic [Logistic Regression Convergence Options](#) for more information.

Output. These options enable you to request additional statistics that will be displayed in the advanced output of the model nugget built by the node. See the topic [Logistic Regression Advanced Output](#) for more information.

Stepping. These options enable you to control the criteria for adding and removing fields with the Stepwise, Forwards, Backwards, or Backwards Stepwise estimation methods. (The button is disabled if the Enter method is selected.) See the topic [Logistic Regression Stepping Options](#) for more information.

Related information

- [Logistic Node](#)
 - [Logistic Node Model Options](#)
 - [Adding Terms to a Logistic Regression Model](#)
 - [Logistic Regression Convergence Options](#)
 - [Logistic Regression Advanced Output](#)
 - [Logistic Regression Stepping Options](#)
-

Logistic Regression Convergence Options

You can set the convergence parameters for logistic regression model estimation.

Maximum iterations. Specify the maximum number of iterations for estimating the model.

Maximum step-halving. Step-halving is a technique used by logistic regression to deal with complexities in the estimation process. Under normal circumstances, you should use the default setting.

Log-likelihood convergence. Iterations stop if the relative change in the log-likelihood is less than this value. The criterion is not used if the value is 0.

Parameter convergence. Iterations stop if the absolute change or relative change in the parameter estimates is less than this value. The criterion is not used if the value is 0.

Delta (Multinomial models only). You can specify a value between 0 and 1 to be added to each empty cell (combination of input field and output field values). This can help the estimation algorithm deal with data where there are many possible combinations of field values relative to the number of records in the data. The default is 0.

Related information

- [Logistic Node](#)
 - [Logistic Node Model Options](#)
 - [Adding Terms to a Logistic Regression Model](#)
 - [Logistic Node Expert Options](#)
 - [Logistic Regression Advanced Output](#)
 - [Logistic Regression Stepping Options](#)
-

Logistic Regression Advanced Output

Select the optional output you want to display in the advanced output of the Regression model nugget. To view the advanced output, browse the model nugget and click the Advanced tab. See the topic [Logistic Model Nugget Advanced Output](#) for more information.

Binomial Options

Select the types of output to be generated for the model. See the topic [Logistic Model Nugget Advanced Output](#) for more information.

Display. Select whether to display the results at each step, or to wait until all steps have been worked through.

CI for exp(B). Select the confidence intervals for each coefficient (shown as Beta) in the expression. Specify the level of the confidence interval (the default is 95%).

Residual Diagnosis. Requests a Casewise Diagnostics table of residuals.

- **Outliers outside (std. dev.).** List only residual cases for which the absolute standardized value of the listed variable is at least as large as the value you specify. The default value is 2.
- **All cases.** Include all cases in the Casewise Diagnostic table of residuals.
Note: Because this option lists each of the input records, it may result in an exceptionally large table in the report, with one line for every record.

Classification cutoff. This enables you to determine the cutpoint for classifying cases. Cases with predicted values that exceed the classification cutoff are classified as positive, while those with predicted values smaller than the cutoff are classified as negative. To change the default, enter a value between 0.01 and 0.99.

Multinomial Options

Select the types of output to be generated for the model. See the topic [Logistic Model Nugget Advanced Output](#) for more information.

Note: Selecting the Likelihood ratio tests option greatly increases the processing time required to build a logistic regression model. If your model is taking too long to build, consider disabling this option or utilize the Wald and Score statistics instead. See the topic [Logistic Regression Stepping Options](#) for more information.

Iteration history for every. Select the step interval for printing iteration status in the advanced output.

Confidence Interval. The confidence intervals for coefficients in the equations. Specify the level of the confidence interval (the default is 95%).

Related information

- [Logistic Node](#)
 - [Logistic Node Model Options](#)
 - [Adding Terms to a Logistic Regression Model](#)
 - [Logistic Node Expert Options](#)
 - [Logistic Regression Convergence Options](#)
 - [Logistic Regression Stepping Options](#)
-

Logistic Regression Stepping Options

These options enable you to control the criteria for adding and removing fields with the Stepwise, Forwards, Backwards, or Backwards Stepwise estimation methods.

Number of terms in model (Multinomial models only). You can specify the minimum number of terms in the model for Backwards and Backwards Stepwise models and the maximum number of terms for Forwards and Stepwise models. If you specify a minimum value greater than 0, the model will include that many terms, even if some of the terms would have been removed based on statistical criteria. The minimum setting is ignored for Forwards, Stepwise, and Enter models. If you specify a maximum, some terms may be omitted from the model, even though they would have been selected based on statistical criteria. The Specify Maximum setting is ignored for Backwards, Backwards Stepwise, and Enter models.

Entry criterion (Multinomial models only). Select Score to maximize speed of processing. The Likelihood Ratio option may provide somewhat more robust estimates but take longer to compute. The default setting is to use the Score statistic.

Removal criterion. Select Likelihood Ratio for a more robust model. To shorten the time required to build the model, you can try selecting Wald. However, if you have complete or quasi-complete separation in the data (which you can determine by using the Advanced tab on the model nugget), the Wald statistic becomes particularly unreliable and should not be used. The default setting is to use the likelihood-ratio statistic. For binomial models, there is the additional option Conditional. This provides removal testing based on the probability of the likelihood-ratio statistic based on conditional parameter estimates.

Significance thresholds for criteria. This option enables you to specify selection criteria based on the statistical probability (the *p* value) associated with each field. Fields will be added to the model only if the associated *p* value is smaller than the Entry value and will be removed only if the *p* value is larger than the Removal value. The Entry value must be smaller than the Removal value.

Requirements for entry or removal (Multinomial models only). For some applications, it doesn't make mathematical sense to add interaction terms to the model unless the model also contains the lower-order terms for the fields involved in the interaction term. For example, it may not make sense to include $A * B$ in the model unless A and B are also included in the model. These options let you determine how such dependencies are handled during stepwise term selection.

- **Hierarchy for discrete effects.** Higher-order effects (interactions involving more fields) will enter the model only if all lower-order effects (main effects or interactions involving fewer fields) for the relevant fields are already in the model, and lower-order effects will not be removed if higher-order effects involving the same fields are in the model. This option applies only to categorical fields.
- **Hierarchy for all effects.** This option works in the same way as the previous option, except that it applies to all input fields.
- **Containment for all effects.** Effects can be included in the model only if all of the effects contained in the effect are also included in the model. This option is similar to the Hierarchy for all effects option except that continuous fields are treated somewhat differently. For an effect to contain another effect, the contained (lower-order) effect must include *all* of the continuous fields involved in the containing (higher-order) effect, and the contained effect's categorical fields must be a subset of those in the containing effect. For example, if A and B are categorical fields and X is a continuous field, the term $A * B * X$ contains the terms $A * X$ and $B * X$.
- **None.** No relationships are enforced; terms are added to and removed from the model independently.

Related information

- [Logistic Node](#)
- [Logistic Node Model Options](#)
- [Adding Terms to a Logistic Regression Model](#)
- [Logistic Node Expert Options](#)
- [Logistic Regression Convergence Options](#)

- [Logistic Regression Advanced Output](#)

Logistic Model Nugget

A Logistic model nugget represents the equation estimated by a Logistic node. It contains all of the information captured by the logistic regression model, as well as information about the model structure and performance. This type of equation may also be generated by other models such as Oracle SVM.

When you run a stream containing a Logistic model nugget, the node adds two new fields containing the model's prediction and the associated probability. The names of the new fields are derived from the name of the output field being predicted, prefixed with $\$L-$ for the predicted category and $\$LP-$ for the associated probability. For example, for an output field named *colorpref*, the new fields would be named $\$L-colorpref$ and $\$LP-colorpref$. In addition, if you have selected the Append all probabilities option in the Logistic node, an additional field will be added for each category of the output field, containing the probability belonging to the corresponding category for each record. These additional fields are named based on the values of the output field, prefixed by $\$LP-$. For example, if the legal values of *colorpref* are *Red*, *Green*, and *Blue*, three new fields will be added: $\$LP-Red$, $\$LP-Green$, and $\$LP-Blue$.

Generating a Filter node. The Generate menu enables you to create a new Filter node to pass input fields based on the results of the model. Fields that are dropped from the model due to multicollinearity will be filtered by the generated node, as well as fields not used in the model.

- [Logistic Nugget Model Details](#)
- [Logistic Model Nugget Summary](#)
- [Logistic Model Nugget Settings](#)
- [Logistic Model Nugget Advanced Output](#)

Related information

- [Logistic Model Nugget Summary](#)
- [Logistic Model Nugget Settings](#)
- [Logistic Model Nugget Advanced Output](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)
- [Logistic Node](#)

Logistic Nugget Model Details

For multinomial models, the Model tab in a Logistic model nugget has a split display with model equations in the left pane, and predictor importance on the right. For binomial models, the tab displays predictor importance only. See the topic [Predictor Importance](#) for more information.

Model Equations

For multinomial models, the left pane displays the actual equations estimated for the logistic regression model. There is one equation for each category in the target field, except the baseline category. The equations are displayed in a tree format. This type of equation may also be generated by certain other models such as Oracle SVM.

Equation For. Shows the regression equations used to derive the target category probabilities, given a set of predictor values. The last category of the target field is considered the **baseline category**; the equations shown give the log-odds for the other target categories relative to the baseline category for a particular set of predictor values. The predicted probability for each category of the given predictor pattern is derived from these log-odds values.

How Probabilities Are Calculated

Each equation calculates the log-odds for a particular target category, relative to the baseline category. The **log-odds**, also called the **logit**, is the ratio of the probability for the specified target category to that of the baseline category, with the natural logarithm function applied to the result. For the baseline category, the odds of the category relative to itself is 1.0, and thus the log-odds is 0. You can think of this as an implicit equation for the baseline category where all coefficients are 0.

To derive the probability from the log-odds for a particular target category, you take the logit value calculated by the equation for that category and apply the following formula:

$$P(\text{group}_i) = \exp(g_i) / \sum_k \exp(g_k)$$

where g is the calculated log-odds, i is the category index, and k goes from 1 to the number of target categories.

Predictor Importance

Optionally, a chart that indicates the relative importance of each predictor in estimating the model may also be displayed on the Model tab. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least. Note this chart is only available if Calculate predictor importance is selected on the Analyze tab before generating the model. See the topic [Predictor Importance](#) for more information.

Note: Predictor importance may take longer to calculate for logistic regression than for other types of models, and is not selected on the Analyze tab by default. Selecting this option may slow performance, particularly with large datasets.

Logistic Model Nugget Summary

The summary for a logistic regression model displays the fields and settings used to generate the model. In addition, if you have executed an Analysis node attached to this modeling node, information from that analysis will also be displayed in this section. For general information on using the model browser, see [Browsing model nuggets](#).

Related information

- [Logistic Model Nugget](#)
 - [Logistic Model Nugget Settings](#)
 - [Logistic Model Nugget Advanced Output](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Logistic Model Nugget Settings

The Settings tab in a Logistic model nugget specifies options for confidences, probabilities, propensity scores, and SQL generation during model scoring. This tab is only available after the model nugget has been added to a stream and displays different options depending on the type of model and target.

Multinomial Models

For multinomial models, the following options are available.

Calculate confidences Specifies whether confidences are calculated during scoring.

Calculate raw propensity scores (flag targets only) For models with flag targets only, you can request raw propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to standard prediction and confidence values. Adjusted propensity scores are not available. See the topic [Modeling Node Analyze Options](#) for more information.

Append all probabilities Specifies whether probabilities for each category of the output field are added to each record processed by the node. If this option is not selected, the probability of only the predicted category is added. For a nominal target with three categories, for example, the scoring output will include a column for each of the three categories, plus a fourth column indicating the probability for whichever category is predicted. For example if the probabilities for categories *Red*, *Green*, and *Blue* are 0.6, 0.3, and 0.1 respectively, the predicted category would be *Red*, with a probability of 0.6.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score by converting to native SQL If selected, generates native SQL to score the model within the database.
Note: Although this option can provide quicker results, the size and complexity of the native SQL increases as the complexity of the model increases.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Note: For multinomial models, SQL generation is unavailable if Append all probabilities has been selected, or—for models with nominal targets—if Calculate confidences has been selected. SQL generation with confidence calculations is supported for multinomial models with flag targets only. SQL generation is not available for binomial models.

Binomial Models

For binomial models, confidences and probabilities are always enabled, and the settings that would enable you to disable these options are not available. SQL generation is not available for binomial models. The only setting that can be changed for binomial models is the ability to calculate raw propensity scores. As noted earlier for multinomial models, this applies to models with flag targets only. See the topic [Modeling Node Analyze Options](#) for more information.

Related information

- [Logistic Model Nugget](#)
 - [Logistic Model Nugget Summary](#)
 - [Logistic Model Nugget Advanced Output](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Logistic Model Nugget Advanced Output

The advanced output for logistic regression (also known as **nominal regression**) gives detailed information about the estimated model and its performance. Most of the information contained in the advanced output is quite technical, and extensive knowledge of logistic regression analysis is required to properly interpret this output.

Warnings. Indicates any warnings or potential problems with the results.

Case processing summary. Lists the number of records processed, broken down by each symbolic field in the model.

Step summary (optional). Lists the effects added or removed at each step of model creation, when using automatic field selection.

Note: Only shown for the Stepwise, Forwards, Backwards, or Backwards Stepwise methods.

Iteration history (optional). Shows the iteration history of parameter estimates for every n iterations beginning with the initial estimates, where n is the value of the print interval. The default is to print every iteration ($n=1$).

Model fitting information (Multinomial models). Shows the likelihood-ratio test of your model (Final) against one in which all of the parameter coefficients are 0 (Intercept Only).

Classification (optional). Shows the matrix of predicted and actual output field values with percentages.

Goodness-of-fit chi-square statistics (optional). Shows Pearson's and likelihood-ratio chi-square statistics. These statistics test the overall fit of the model to the training data.

Hosmer and Lemeshow goodness-of-fit (optional). Shows the results of grouping cases into deciles of risk and comparing the observed probability with the expected probability within each decile. This goodness-of-fit statistic is more robust than the traditional goodness-of-fit statistic used in multinomial models, particularly for models with continuous covariates and studies with small sample sizes.

Pseudo R-square (optional). Shows the Cox and Snell, Nagelkerke, and McFadden R -square measures of model fit. These statistics are in some ways analogous to the R -square statistic in linear regression.

Monotonicity measures (optional). Shows the number of concordant pairs, discordant pairs, and tied pairs in the data, as well as the percentage of the total number of pairs that each represents. The Somers' D, Goodman and Kruskal's Gamma, Kendall's tau-a, and Concordance Index C are also displayed in this table.

Information criteria (optional). Shows Akaike's information criterion (AIC) and Schwarz's Bayesian information criterion (BIC).

Likelihood ratio tests (optional). Shows statistics testing of whether the coefficients of the model effects are statistically different from 0. Significant input fields are those with very small significance levels in the output (labeled *Sig.*).

Parameter estimates (optional). Shows estimates of the equation coefficients, tests of those coefficients, odds ratios derived from the coefficients labeled *Exp(B)*, and confidence intervals for the odds ratios.

Asymptotic covariance/correlation matrix (optional). Shows the asymptotic covariances and/or correlations of the coefficient estimates.

Observed and predicted frequencies (optional). For each covariate pattern, shows the observed and predicted frequencies for each output field value. This table can be quite large, especially for models with numeric input fields. If the resulting table would be too large to be practical, it is omitted, and a warning is displayed.

Related information

- [Logistic Model Nugget](#)
 - [Logistic Model Nugget Summary](#)
 - [Logistic Model Nugget Settings](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

PCA/Factor Node

The PCA/Factor node provides powerful data-reduction techniques to reduce the complexity of your data. Two similar but distinct approaches are provided.

- **Principal components analysis (PCA)** finds linear combinations of the input fields that do the best job of capturing the variance in the entire set of fields, where the components are orthogonal (perpendicular) to each other. PCA focuses on all variance, including both shared and unique variance.
- **Factor analysis** attempts to identify underlying concepts, or **factors**, that explain the pattern of correlations within a set of observed fields. Factor analysis focuses on shared variance only. Variance that is unique to specific fields is not considered in estimating the model. Several methods of factor analysis are provided by the Factor/PCA node.

For both approaches, the goal is to find a small number of derived fields that effectively summarize the information in the original set of fields.

Requirements. Only numeric fields can be used in a PCA-Factor model. To estimate a factor analysis or PCA, you need one or more fields with the role set to *Input* fields. Fields with the role set to *Target*, *Both*, or *None* are ignored, as are non-numeric fields.

Strengths. Factor analysis and PCA can effectively reduce the complexity of your data without sacrificing much of the information content. These techniques can help you build more robust models that execute more quickly than would be possible with the raw input fields.

- [PCA/Factor Node Model Options](#)
- [PCA/Factor Node Expert Options](#)
- [PCA/Factor Node Rotation Options](#)

Related information

- [Overview of modeling nodes](#)
 - [Modeling Node Fields Options](#)
 - [PCA/Factor Node Model Options](#)
 - [PCA/Factor Node Expert Options](#)
 - [PCA/Factor Node Rotation Options](#)
-

PCA/Factor Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Extraction Method. Specify the method to be used for data reduction.

- **Principal Components.** This is the default method, which uses PCA to find components that summarize the input fields.
- **Unweighted Least Squares.** This factor analysis method works by finding the set of factors that is best able to reproduce the pattern of relationships (correlations) among the input fields.
- **Generalized Least Squares.** This factor analysis method is similar to unweighted least squares, except that it uses weighting to de-emphasize fields with a lot of unique (unshared) variance.
- **Maximum Likelihood.** This factor analysis method produces factor equations that are most likely to have produced the observed pattern of relationships (correlations) in the input fields, based on assumptions about the form of those relationships.
- **Principal Axis Factoring.** This factor analysis method is very similar to the principal components method, except that it focuses on shared variance only.
- **Alpha Factoring.** This factor analysis method considers the fields in the analysis to be a sample from the universe of potential input fields. It maximizes the statistical reliability of the factors.
- **Image Factoring.** This factor analysis method uses data estimation to isolate the common variance and find factors that describe it.

Related information

- [PCA/Factor Node](#)
 - [PCA/Factor Node Expert Options](#)
 - [PCA/Factor Node Rotation Options](#)
-

PCA/Factor Node Expert Options

If you have detailed knowledge of factor analysis and PCA, expert options enable you to fine-tune the training process. To access expert options, set Mode to Expert on the Expert tab.

Missing values. By default, IBM® SPSS® Modeler only uses records that have valid values for all fields used in the model. (This is sometimes called **listwise deletion** of missing values.) If you have a lot of missing data, you may find that this approach eliminates too many records, leaving you without enough data to generate a good model. In such cases, you can deselect the Only use complete records option. IBM SPSS Modeler then attempts to use as much information as possible to estimate the model, including records where some of the fields have missing values. (This is sometimes called **pairwise deletion** of missing values.) However, in some situations, using incomplete records in this manner can lead to computational problems in estimating the model.

Fields. Specify whether to use the correlation matrix (the default) or the covariance matrix of the input fields in estimating the model.

Maximum iterations for convergence. Specify the maximum number of iterations for estimating the model.

Extract factors. There are two ways to select the number of factors to extract from the input fields.

- **Eigenvalues over.** This option will retain all factors or components with eigenvalues larger than the specified criterion. **Eigenvalues** measure the ability of each factor or component to summarize variance in the set of input fields. The model will retain all factors or components with eigenvalues greater than the specified value when using the correlation matrix. When using the covariance matrix, the criterion is the specified value times the mean eigenvalue. That scaling gives this option a similar meaning for both types of matrix.
- **Maximum number.** This option will retain the specified number of factors or components in descending order of eigenvalues. In other words, the factors or components corresponding to the n highest eigenvalues are retained, where n is the specified criterion. The default extraction criterion is five factors/components.

Component/factor matrix format. These options control the format of the factor matrix (or component matrix for PCA models).

- **Sort values.** If this option is selected, factor loadings in the model output will be sorted numerically.
- **Hide values below.** If this option is selected, scores below the specified threshold will be hidden in the matrix to make it easier to see the pattern in the matrix.

Rotation. These options enable you to control the rotation method for the model. See the topic [PCA/Factor Node Rotation Options](#) for more information.

Related information

- [PCA/Factor Node](#)
 - [PCA/Factor Node Model Options](#)
 - [PCA/Factor Node Rotation Options](#)
-

PCA/Factor Node Rotation Options

In many cases, mathematically rotating the set of retained factors can increase their usefulness and especially their interpretability. Select a rotation method:

- **No rotation.** The default option. No rotation is used.
- **Varimax.** An orthogonal rotation method that minimizes the number of fields with high loadings on each factor. It simplifies the interpretation of the factors.
- **Direct oblimin.** A method for oblique (non-orthogonal) rotation. When Delta equals 0 (the default), solutions are oblique. As delta becomes more negative, the factors become less oblique. To override the default delta of 0, enter a number less than or equal to 0.8.
- **Quartimax.** An orthogonal method that minimizes the number of factors needed to explain each field. It simplifies the interpretation of the observed fields.
- **Equamax.** A rotation method that is a combination of the Varimax method, which simplifies the factors, and the Quartimax method, which simplifies the fields. The number of fields that load highly on a factor and the number of factors needed to explain a field are minimized.
- **Promax.** An oblique rotation, which enables factors to be correlated. It can be calculated more quickly than a direct oblimin rotation, so it can be useful for large datasets. Kappa controls the obliqueness of the solution (the extent to which factors can be correlated).

Related information

- [PCA/Factor Node](#)
 - [PCA/Factor Node Model Options](#)
 - [PCA/Factor Node Expert Options](#)
-

PCA/Factor Model Nugget

A PCA/Factor model nugget represents the factor analysis and principal component analysis (PCA) model created by a PCA/Factor node. They contain all of the information captured by the trained model, as well as information about the model's performance and characteristics.

When you run a stream containing a factor equation model, the node adds a new field for each factor or component in the model. The new field names are derived from the model name, prefixed by $\$F\text{-}$ and suffixed by $-n$, where n is the number of the factor or component. For example, if your model is named *Factor* and contains three factors, the new fields would be named $\$F\text{-Factor-1}$, $\$F\text{-Factor-2}$, and $\$F\text{-Factor-3}$.

To get a better sense of what the factor model has encoded, you can do some more downstream analysis. A useful way to view the result of the factor model is to view the correlations between factors and input fields using a Statistics node. This shows you which input fields load heavily on which factors and can help you discover if your factors have any underlying meaning or interpretation.

You can also assess the factor model by using the information available in the advanced output. To view the advanced output, click the Advanced tab of the model nugget browser. The advanced output contains a lot of detailed information and is meant for users with extensive knowledge of factor analysis or PCA. See the topic [PCA/Factor Model Nugget Advanced Output](#) for more information.

- [PCA/Factor Model Nugget Equations](#)
- [PCA/Factor Model Nugget Summary](#)
- [PCA/Factor Model Nugget Advanced Output](#)

Related information

- [PCA/Factor Model Nugget Equations](#)
 - [PCA/Factor Model Nugget Summary](#)
 - [PCA/Factor Model Nugget Advanced Output](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
 - [PCA/Factor Node](#)
-

PCA/Factor Model Nugget Equations

The Model tab for a Factor model nugget displays the factor score equation for each factor. Factor or component scores are calculated by multiplying each input field value by its coefficient and summing the results.

Related information

- [PCA/Factor Model Nugget](#)
 - [PCA/Factor Model Nugget Summary](#)
 - [PCA/Factor Model Nugget Advanced Output](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

PCA/Factor Model Nugget Summary

The Summary tab for a factor model displays the number of factors retained in the factor/PCA model, along with additional information on the fields and settings used to generate the model. See the topic [Browsing model nuggets](#) for more information.

Related information

- [PCA/Factor Model Nugget](#)

- [PCA/Factor Model Nugget Equations](#)
 - [PCA/Factor Model Nugget Advanced Output](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

PCA/Factor Model Nugget Advanced Output

The advanced output for factor analysis gives detailed information on the estimated model and its performance. Most of the information contained in the advanced output is quite technical, and extensive knowledge of factor analysis is required to properly interpret this output.

Warnings. Indicates any warnings or potential problems with the results.

Communalities. Shows the proportion of each field's variance that is accounted for by the factors or components. *Initial* gives the initial communalities with the full set of factors (the model starts with as many factors as input fields), and *Extraction* gives the communalities based on the retained set of factors.

Total variance explained. Shows the total variance explained by the factors in the model. *Initial Eigenvalues* shows the variance explained by the full set of initial factors. *Extraction Sums of Squared Loadings* shows the variance explained by factors retained in the model. *Rotation Sums of Squared Loadings* shows the variance explained by the rotated factors. Note that for oblique rotations, *Rotation Sums of Squared Loadings* shows only the sums of squared loadings and does not show variance percentages.

Factor (or component) matrix. Shows correlations between input fields and unrotated factors.

Rotated factor (or component) matrix. Shows correlations between input fields and rotated factors for orthogonal rotations.

Pattern matrix. Shows the partial correlations between input fields and rotated factors for oblique rotations.

Structure matrix. Shows the simple correlations between input fields and rotated factors for oblique rotations.

Factor correlation matrix. Shows correlations among factors for oblique rotations.

Related information

- [PCA/Factor Model Nugget](#)
 - [PCA/Factor Model Nugget Equations](#)
 - [PCA/Factor Model Nugget Summary](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Discriminant node

Discriminant analysis builds a predictive model for group membership. The model is composed of a discriminant function (or, for more than two groups, a set of discriminant functions) based on linear combinations of the predictor variables that provide the best discrimination between the groups. The functions are generated from a sample of cases for which group membership is known; the functions can then be applied to new cases that have measurements for the predictor variables but have unknown group membership.

Example. A telecommunications company can use discriminant analysis to classify customers into groups based on usage data. This allows them to score potential customers and target those who are most likely to be in the most valuable groups.

Requirements. You need one or more input fields and exactly one target field. The target must be a categorical field (with a measurement level of *Flag* or *Nominal*) with string or integer storage. (Storage can be converted using a Filler or Derive node if necessary.) Fields set to *Both* or *None* are ignored. Fields used in the model must have their types fully instantiated.

Strengths. Discriminant analysis and Logistic Regression are both suitable classification models. However, Discriminant analysis makes more assumptions about the input fields—for example, they are normally distributed and should be continuous, and they give better results if those requirements are met, especially if the sample size is small.

- [Discriminant Node Model Options](#)
- [Discriminant Node Expert Options](#)
- [Discriminant Node Output Options](#)
- [Discriminant Node Stepping Options](#)

- [Discriminant Model Nugget](#)
-

Discriminant Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Create split models. Builds a separate model for each possible value of input fields that are specified as split fields. See [Building Split Models](#) for more information.

Method. The following options are available for entering predictors into the model:

- **Enter.** This is the default method, which enters all of the terms into the equation directly. Terms that do not add significantly to the predictive power of the model are not added.
- **Stepwise.** The initial model is the simplest model possible, with no model terms (except the constant) in the equation. At each step, terms that have not yet been added to the model are evaluated, and if the best of those terms adds significantly to the predictive power of the model, it is added.

Note: The Stepwise method has a strong tendency to overfit the training data. When using these methods, it is especially important to verify the validity of the resulting model with a hold-out test sample or new data.

Related information

- [Discriminant node](#)
 - [Discriminant Node Expert Options](#)
 - [Discriminant Node Output Options](#)
 - [Discriminant Node Stepping Options](#)
-

Discriminant Node Expert Options

If you have detailed knowledge of discriminant analysis, expert options allow you to fine-tune the training process. To access expert options, set Mode to Expert on the Expert tab.

Prior Probabilities. This option determines whether the classification coefficients are adjusted for a priori knowledge of group membership.

- **All groups equal.** Equal prior probabilities are assumed for all groups; this has no effect on the coefficients.
- **Compute from group sizes.** The observed group sizes in your sample determine the prior probabilities of group membership. For example, if 50% of the observations included in the analysis fall into the first group, 25% in the second, and 25% in the third, the classification coefficients are adjusted to increase the likelihood of membership in the first group relative to the other two.

Use Covariance Matrix. You can choose to classify cases using a within-groups covariance matrix or a separate-groups covariance matrix.

- *Within-groups.* The pooled within-groups covariance matrix is used to classify cases.
- *Separate-groups.* Separate-groups covariance matrices are used for classification. Because classification is based on the discriminant functions (not based on the original variables), this option is not always equivalent to quadratic discrimination.

Output. These options allow you to request additional statistics that will be displayed in the advanced output of the model nugget built by the node. See the topic [Discriminant Node Output Options](#) for more information.

Stepping. These options allow you to control the criteria for adding and removing fields with the Stepwise estimation method. (The button is disabled if the Enter method is selected.) See the topic [Discriminant Node Stepping Options](#) for more information.

Related information

- [Discriminant node](#)
 - [Discriminant Node Model Options](#)
 - [Discriminant Node Output Options](#)
 - [Discriminant Node Stepping Options](#)
-

Discriminant Node Output Options

Select the optional output you want to display in the advanced output of the logistic regression model nugget. To view the advanced output, browse the model nugget and click the Advanced tab. See the topic [Discriminant Model Nugget Advanced Output](#) for more information.

Descriptives. Available options are means (including standard deviations), univariate ANOVAs, and Box's M test.

- *Means.* Displays total and group means, as well as standard deviations for the independent variables.
- *Univariate ANOVAs.* Performs a one-way analysis-of-variance test for equality of group means for each independent variable.
- *Box's M .* A test for the equality of the group covariance matrices. For sufficiently large samples, a nonsignificant p value means there is insufficient evidence that the matrices differ. The test is sensitive to departures from multivariate normality.

Function Coefficients. Available options are Fisher's classification coefficients and unstandardized coefficients.

- *Fisher's.* Displays Fisher's classification function coefficients that can be used directly for classification. A separate set of classification function coefficients is obtained for each group, and a case is assigned to the group for which it has the largest discriminant score (classification function value).
- *Unstandardized.* Displays the unstandardized discriminant function coefficients.

Matrices. Available matrices of coefficients for independent variables are within-groups correlation matrix, within-groups covariance matrix, separate-groups covariance matrix, and total covariance matrix.

- *Within-groups correlation.* Displays a pooled within-groups correlation matrix that is obtained by averaging the separate covariance matrices for all groups before computing the correlations.
- *Within-groups covariance.* Displays a pooled within-groups covariance matrix, which may differ from the total covariance matrix. The matrix is obtained by averaging the separate covariance matrices for all groups.
- *Separate-groups covariance.* Displays separate covariance matrices for each group.
- *Total covariance.* Displays a covariance matrix from all cases as if they were from a single sample.

Classification. The following output pertains to the classification results.

- *Casewise results.* Codes for actual group, predicted group, posterior probabilities, and discriminant scores are displayed for each case.
- *Summary table.* The number of cases correctly and incorrectly assigned to each of the groups based on the discriminant analysis. Sometimes called the "Confusion Matrix."
- *Leave-one-out classification.* Each case in the analysis is classified by the functions derived from all cases other than that case. It is also known as the "U-method."
- *Territorial map.* A plot of the boundaries used to classify cases into groups based on function values. The numbers correspond to groups into which cases are classified. The mean for each group is indicated by an asterisk within its boundaries. The map is not displayed if there is only one discriminant function.
- *Combined-groups.* Creates an all-groups scatterplot of the first two discriminant function values. If there is only one function, a histogram is displayed instead.
- *Separate-groups.* Creates separate-group scatterplots of the first two discriminant function values. If there is only one function, histograms are displayed instead.

Stepwise. Summary of Steps displays statistics for all variables after each step; F for pairwise distances displays a matrix of pairwise F ratios for each pair of groups. The F ratios can be used for significance tests of the Mahalanobis distances between groups.

Related information

- [Discriminant node](#)
 - [Discriminant Node Model Options](#)
 - [Discriminant Node Expert Options](#)
 - [Discriminant Node Stepping Options](#)
-

Discriminant Node Stepping Options

These options allow you to control the method and criteria for adding fields with the Stepwise estimation method.

Method. Select the statistic to be used for entering or removing new variables. Available alternatives are Wilks' lambda, unexplained variance, Mahalanobis distance, smallest F ratio, and Rao's V . With Rao's V , you can specify the minimum increase in V for a variable to enter.

- *Wilks' lambda.* A variable selection method for stepwise discriminant analysis that chooses variables for entry into the equation on the basis of how much they lower Wilks' lambda. At each step, the variable that minimizes the overall Wilks' lambda is entered.
- *Unexplained variance.* At each step, the variable that minimizes the sum of the unexplained variation between groups is entered.
- *Mahalanobis distance.* A measure of how much a case's values on the independent variables differ from the average of all cases. A large Mahalanobis distance identifies a case as having extreme values on one or more of the independent variables.
- *Smallest F ratio.* A method of variable selection in stepwise analysis based on maximizing an F ratio computed from the Mahalanobis distance between groups.
- *Rao's V .* A measure of the differences between group means. Also called the Lawley-Hotelling trace. At each step, the variable that maximizes the increase in Rao's V is entered. After selecting this option, enter the minimum value a variable must have to enter the

analysis.

Criteria. Available alternatives are Use F value and Use probability of F. Enter values for entering and removing variables.

- **Use F value.** A variable is entered into the model if its F value is greater than the Entry value and is removed if the F value is less than the Removal value. Entry must be greater than Removal, and both values must be positive. To enter more variables into the model, lower the Entry value. To remove more variables from the model, increase the Removal value.
- **Use probability of F.** A variable is entered into the model if the significance level of its F value is less than the Entry value and is removed if the significance level is greater than the Removal value. Entry must be less than Removal, and both values must be positive. To enter more variables into the model, increase the Entry value. To remove more variables from the model, lower the Removal value.

Related information

- [Discriminant node](#)
- [Discriminant Node Model Options](#)
- [Discriminant Node Expert Options](#)
- [Discriminant Node Output Options](#)

Discriminant Model Nugget

Discriminant model nuggets represent the equations estimated by Discriminant nodes. They contain all of the information captured by the discriminant model, as well as information about the model structure and performance.

When you run a stream containing a Discriminant model nugget, the node adds two new fields containing the model's prediction and the associated probability. The names of the new fields are derived from the name of the output field being predicted, prefixed with \$D- for the predicted category and \$DP- for the associated probability. For example, for an output field named *colorpref*, the new fields would be named \$D-colorpref and \$DP-colorpref.

Generating a Filter node. The Generate menu allows you to create a new Filter node to pass input fields based on the results of the model.

Predictor Importance

Optionally, a chart that indicates the relative importance of each predictor in estimating the model may also be displayed on the Model tab. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least. Note this chart is only available if Calculate predictor importance is selected on the Analyze tab before generating the model. See the topic [Predictor Importance](#) for more information.

- [Discriminant Model Nugget Advanced Output](#)
- [Discriminant Model Nugget Settings](#)
- [Discriminant Model Nugget Summary](#)

Related information

- [Discriminant Model Nugget Advanced Output](#)
- [Discriminant Model Nugget Settings](#)
- [Discriminant Model Nugget Summary](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)
- [Logistic Node](#)

Discriminant Model Nugget Advanced Output

The advanced output for discriminant analysis gives detailed information about the estimated model and its performance. Most of the information contained in the advanced output is quite technical, and extensive knowledge of discriminant analysis is required to properly interpret this output. See the topic [Discriminant Node Output Options](#) for more information.

Related information

- [Discriminant Model Nugget](#)
- [Discriminant Model Nugget Settings](#)
- [Discriminant Model Nugget Summary](#)

- [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Discriminant Model Nugget Settings

The Settings tab in a Discriminant model nugget allows you to obtain propensity scores when scoring the model. This tab is available for models with flag targets only, and only after the model nugget has been added to a stream.

Calculate raw propensity scores. For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

Calculate adjusted propensity scores. Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Related information

- [Discriminant Model Nugget](#)
 - [Discriminant Model Nugget Advanced Output](#)
 - [Discriminant Model Nugget Summary](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Discriminant Model Nugget Summary

The Summary tab for a Discriminant model nugget displays the fields and settings used to generate the model. In addition, if you have executed an Analysis node attached to this modeling node, information from that analysis will also be displayed in this section. For general information on using the model browser, see [Browsing model nuggets](#).

Related information

- [Discriminant Model Nugget](#)
 - [Discriminant Model Nugget Advanced Output](#)
 - [Discriminant Model Nugget Settings](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

GenLin Node

The generalized linear model expands the general linear model so that the dependent variable is linearly related to the factors and covariates via a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers widely used statistical models, such as linear regression for normally distributed responses, logistic models for binary data, loglinear models for count data,

complementary log-log models for interval-censored survival data, plus many other statistical models through its very general model formulation.

Examples. A shipping company can use generalized linear models to fit a Poisson regression to damage counts for several types of ships constructed in different time periods, and the resulting model can help determine which ship types are most prone to damage.

A car insurance company can use generalized linear models to fit a gamma regression to damage claims for cars, and the resulting model can help determine the factors that contribute the most to claim size.

Medical researchers can use generalized linear models to fit a complementary log-log regression to interval-censored survival data to predict the time to recurrence for a medical condition.

Generalized linear models work by building an equation that relates the input field values to the output field values. Once the model is generated, it can be used to estimate values for new data. For each record, a probability of membership is computed for each possible output category. The target category with the highest probability is assigned as the predicted output value for that record.

Requirements. You need one or more input fields and exactly one target field (which can have a measurement level of *Continuous* or *Flag*) with two or more categories. Fields used in the model must have their types fully instantiated.

Strengths. The generalized linear model is extremely flexible, but the process of choosing the model structure is not automated and thus demands a level of familiarity with your data that is not required by "black box" algorithms.

- [GenLin Node Field Options](#)
- [GenLin Node Model Options](#)
- [GenLin Node Expert Options](#)
- [Generalized Linear Models Iterations](#)
- [Generalized Linear Models Advanced Output](#)
- [GenLin Model Nugget](#)

Related information

- [Overview of modeling nodes](#)
- [Modeling Node Fields Options](#)
- [GenLin Node Field Options](#)
- [GenLin Node Model Options](#)
- [GenLin Node Expert Options](#)
- [Generalized Linear Models Iterations](#)
- [Generalized Linear Models Advanced Output](#)

GenLin Node Field Options

In addition to the target, input, and partition custom options typically offered on modeling node Fields tabs (see [Modeling Node Fields Options](#)), the GenLin node offers the following extra functionality.

Use weight field. The scale parameter is an estimated model parameter related to the variance of the response. The scale weights are "known" values that can vary from observation to observation. If the scale weight variable is specified, the scale parameter, which is related to the variance of the response, is divided by it for each observation. Records with scale weight values that are less than or equal to 0 or are missing are not used in the analysis.

Target field represents number of events occurring in a set of trials. When the response is a number of events occurring in a set of trials, the target field contains the number of events and you can select an additional variable containing the number of trials. Alternatively, if the number of trials is the same across all subjects, then trials may be specified using a fixed value. The number of trials should be greater than or equal to the number of events for each record. Events should be non-negative integers, and trials should be positive integers.

Related information

- [GenLin Node](#)
- [GenLin Node Model Options](#)
- [GenLin Node Expert Options](#)
- [Generalized Linear Models Iterations](#)
- [Generalized Linear Models Advanced Output](#)

GenLin Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Create split models. Builds a separate model for each possible value of input fields that are specified as split fields. See [Building Split Models](#) for more information.

Model type. There are two options for the type of model to build. Main effects only causes the model to include only the input fields individually, and not to test interactions (multiplicative effects) between input fields. Main effects and all two-way interactions includes all two-way interactions as well as the input field main effects.

Offset. The offset term is a "structural" predictor. Its coefficient is not estimated by the model but is assumed to have the value 1; thus, the values of the offset are simply added to the linear predictor of the target. This is especially useful in Poisson regression models, where each case may have different levels of exposure to the event of interest.

For example, when modeling accident rates for individual drivers, there is an important difference between a driver who has been at fault in one accident in three years of experience and a driver who has been at fault in one accident in 25 years! The number of accidents can be modeled as a Poisson or negative binomial response with a log link if the natural log of the experience of the driver is included as an offset term.

Other combinations of distribution and link types would require other transformations of the offset variable.

Note: If a variable offset field is used, the specified field should not also be used as an input. Set the role for the offset field to None in an upstream source or Type node if necessary.

Base category for flag target.

For binary response, you can choose the reference category for the dependent variable. This can affect certain output, such as parameter estimates and saved values, but it should not change the model fit. For example, if your binary response takes values 0 and 1:

- By default, the procedure makes the last (highest-valued) category, or 1, the reference category. In this situation, model-saved probabilities estimate the chance that a given case takes the value 0, and parameter estimates should be interpreted as relating to the likelihood of category 0.
- If you specify the first (lowest-valued) category, or 0, as the reference category, then model-saved probabilities estimate the chance that a given case takes the value 1.
- If you specify the custom category and your variable has defined labels, you can set the reference category by choosing a value from the list. This can be convenient when, in the middle of specifying a model, you don't remember exactly how a particular variable was coded.

Include intercept in model. The intercept is usually included in the model. If you can assume the data pass through the origin, you can exclude the intercept.

Related information

- [GenLin Node](#)
 - [GenLin Node Field Options](#)
 - [GenLin Node Expert Options](#)
 - [Generalized Linear Models Iterations](#)
 - [Generalized Linear Models Advanced Output](#)
-

GenLin Node Expert Options

If you have detailed knowledge of generalized linear models, expert options allow you to fine-tune the training process. To access expert options, set Mode to Expert on the Expert tab.

Target Field Distribution and Link Function

Distribution.

This selection specifies the distribution of the dependent variable. The ability to specify a non-normal distribution and non-identity link function is the essential improvement of the generalized linear model over the general linear model. There are many possible distribution-link function combinations, and several may be appropriate for any given dataset, so your choice can be guided by a priori theoretical considerations or which combination seems to fit best.

- **Binomial.** This distribution is appropriate only for variables that represent a binary response or number of events.
- **Gamma.** This distribution is appropriate for variables with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.
- **Inverse Gaussian.** This distribution is appropriate for variables with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.

- **Negative binomial.** This distribution can be thought of as the number of trials required to observe k successes and is appropriate for variables with non-negative integer values. If a data value is non-integer, less than 0, or missing, then the corresponding case is not used in the analysis. The fixed value of the negative binomial distribution's ancillary parameter can be any number greater than or equal to 0. When the ancillary parameter is set to 0, using this distribution is equivalent to using the Poisson distribution.
- **Normal.** This is appropriate for scale variables whose values take a symmetric, bell-shaped distribution about a central (mean) value. The dependent variable must be numeric.
- **Poisson.** This distribution can be thought of as the number of occurrences of an event of interest in a fixed period of time and is appropriate for variables with non-negative integer values. If a data value is non-integer, less than 0, or missing, then the corresponding case is not used in the analysis.
- **Tweedie.** This distribution is appropriate for variables that can be represented by Poisson mixtures of gamma distributions; the distribution is "mixed" in the sense that it combines properties of continuous (takes non-negative real values) and discrete distributions (positive probability mass at a single value, 0). The dependent variable must be numeric, with data values greater than or equal to zero. If a data value is less than zero or missing, then the corresponding case is not used in the analysis. The fixed value of the Tweedie distribution's parameter can be any number greater than one and less than two.
- **Multinomial.** This distribution is appropriate for variables that represent an ordinal response. The dependent variable can be numeric or string, and it must have at least two distinct valid data values.

Link Functions.

The link function is a transformation of the dependent variable that allows estimation of the model. The following functions are available:

- **Identity.** $f(x)=x$. The dependent variable is not transformed. This link can be used with any distribution.
- **Complementary log-log.** $f(x)=\log(-\log(1-x))$. This is appropriate only with the binomial distribution.
- **Cumulative Cauchit.** $f(x)=\tan(\pi(x-0.5))$, applied to the cumulative probability of each category of the response. This is appropriate only with the multinomial distribution.
- **Cumulative complementary log-log.** $f(x)=\ln(-\ln(1-x))$, applied to the cumulative probability of each category of the response. This is appropriate only with the multinomial distribution.
- **Cumulative logit.** $f(x)=\ln(x/(1-x))$, applied to the cumulative probability of each category of the response. This is appropriate only with the multinomial distribution.
- **Cumulative negative log-log.** $f(x)=-\ln(-\ln(x))$, applied to the cumulative probability of each category of the response. This is appropriate only with the multinomial distribution.
- **Cumulative probit.** $f(x)=\Phi^{-1}(x)$, applied to the cumulative probability of each category of the response, where Φ^{-1} is the inverse standard normal cumulative distribution function. This is appropriate only with the multinomial distribution.
- **Log.** $f(x)=\log(x)$. This link can be used with any distribution.
- **Log complement.** $f(x)=\log(1-x)$. This is appropriate only with the binomial distribution.
- **Logit.** $f(x)=\log(x/(1-x))$. This is appropriate only with the binomial distribution.
- **Negative binomial.** $f(x)=\log(x/(x+k^{-1}))$, where k is the ancillary parameter of the negative binomial distribution. This is appropriate only with the negative binomial distribution.
- **Negative log-log.** $f(x)=-\log(-\log(x))$. This is appropriate only with the binomial distribution.
- **Odds power.** $f(x)=[(x/(1-x))^{\alpha}-1]/\alpha$, if $\alpha \neq 0$. $f(x)=\log(x)$, if $\alpha=0$. α is the required number specification and must be a real number. This is appropriate only with the binomial distribution.
- **Probit.** $f(x)=\Phi^{-1}(x)$, where Φ^{-1} is the inverse standard normal cumulative distribution function. This is appropriate only with the binomial distribution.
- **Power.** $f(x)=x^{\alpha}$, if $\alpha \neq 0$. $f(x)=\log(x)$, if $\alpha=0$. α is the required number specification and must be a real number. This link can be used with any distribution.

Parameters. The controls in this group allow you to specify parameter values when certain distribution options are chosen.

- **Parameter for negative binomial.** For negative binomial distribution, choose either to specify a value or to allow the system to provide an estimated value.
- **Parameter for Tweedie.** For Tweedie distribution, specify a number between 1.0 and 2.0 for the fixed value.
- **Parameter Estimation.** The controls in this group allow you to specify estimation methods and to provide initial values for the parameter estimates.
 - **Method.** You can select a parameter estimation method. Choose between Newton-Raphson, Fisher scoring, or a hybrid method in which Fisher scoring iterations are performed before switching to the Newton-Raphson method. If convergence is achieved during the Fisher scoring phase of the hybrid method before the maximum number of Fisher iterations is reached, the algorithm continues with the Newton-Raphson method.
 - **Scale Parameter Method.** You can select the scale parameter estimation method. Maximum-likelihood jointly estimates the scale parameter with the model effects; note that this option is not valid if the response has a negative binomial, Poisson, or binomial distribution. The deviance and Pearson chi-square options estimate the scale parameter from the value of those statistics. Alternatively, you can specify a fixed value for the scale parameter.
- **Covariance matrix.** The model-based estimator is the negative of the generalized inverse of the Hessian matrix. The robust (also called the Huber/White/sandwich) estimator is a "corrected" model-based estimator that provides a consistent estimate of the covariance, even when the specification of the variance and link functions is incorrect.

Iterations. These options allow you to control the parameters for model convergence. See the topic [Generalized Linear Models Iterations](#) for more information.

Output. These options allow you to request additional statistics that will be displayed in the advanced output of the model nugget built by the node. See the topic [Generalized Linear Models Advanced Output](#) for more information.

Singularity Tolerance. Singular (or non-invertible) matrices have linearly dependent columns, which can cause serious problems for the estimation algorithm. Even near-singular matrices can lead to poor results, so the procedure will treat a matrix whose determinant is less than the tolerance as singular. Specify a positive value.

Related information

- [GenLin Node](#)
 - [GenLin Node Field Options](#)
 - [GenLin Node Model Options](#)
 - [Generalized Linear Models Iterations](#)
 - [Generalized Linear Models Advanced Output](#)
-

Generalized Linear Models Iterations

You can set the convergence parameters for estimating the generalized linear model.

Iterations. The following options are available:

- **Maximum iterations.** The maximum number of iterations the algorithm will execute. Specify a non-negative integer.
- **Maximum step-halving.** At each iteration, the step size is reduced by a factor of 0.5 until the log-likelihood increases or maximum step-halving is reached. Specify a positive integer.
- **Check for separation of data points.** When selected, the algorithm performs tests to ensure that the parameter estimates have unique values. Separation occurs when the procedure can produce a model that correctly classifies every case. This option is available for binomial responses with binary format .

Convergence Criteria. The following options are available

- **Change in parameter estimates.** When selected, the algorithm stops after an iteration in which the absolute or relative change in the parameter estimates is less than the value specified, which must be positive.
- **Change in log-likelihood.** When selected, the algorithm stops after an iteration in which the absolute or relative change in the log-likelihood function is less than the value specified, which must be positive.
- **Hessian convergence.** For the Absolute specification, convergence is assumed if a statistic based on the Hessian convergence is less than the positive value specified. For the Relative specification, convergence is assumed if the statistic is less than the product of the positive value specified and the absolute value of the log-likelihood.

Related information

- [GenLin Node](#)
 - [GenLin Node Field Options](#)
 - [GenLin Node Model Options](#)
 - [GenLin Node Expert Options](#)
 - [Generalized Linear Models Advanced Output](#)
-

Generalized Linear Models Advanced Output

Select the optional output you want to display in the advanced output of the generalized linear model nugget. To view the advanced output, browse the model nugget and click the Advanced tab. See the topic [GenLin Model Nugget Advanced Output](#) for more information.

The following output is available:

- **Case processing summary.** Displays the number and percentage of cases included and excluded from the analysis and the Correlated Data Summary table.
- **Descriptive statistics.** Displays descriptive statistics and summary information about the dependent variable, covariates, and factors.
- **Model information.** Displays the dataset name, dependent variable or events and trials variables, offset variable, scale weight variable, probability distribution, and link function.
- **Goodness of fit statistics.** Displays deviance and scaled deviance, Pearson chi-square and scaled Pearson chi-square, log-likelihood, Akaike's information criterion (AIC), finite sample corrected AIC (AICC), Bayesian information criterion (BIC), and consistent AIC (CAIC).
- **Model summary statistics.** Displays model fit tests, including likelihood-ratio statistics for the model fit omnibus test and statistics for the Type I or III contrasts for each effect.

- **Parameter estimates.** Displays parameter estimates and corresponding test statistics and confidence intervals. You can optionally display exponentiated parameter estimates in addition to the raw parameter estimates.
- **Covariance matrix for parameter estimates.** Displays the estimated parameter covariance matrix.
- **Correlation matrix for parameter estimates.** Displays the estimated parameter correlation matrix.
- **Contrast coefficient (L) matrices.** Displays contrast coefficients for the default effects and for the estimated marginal means, if requested on the EM Means tab.
- **General estimable functions.** Displays the matrices for generating the contrast coefficient (L) matrices.
- **Iteration history.** Displays the iteration history for the parameter estimates and log-likelihood and prints the last evaluation of the gradient vector and the Hessian matrix. The iteration history table displays parameter estimates for every n^{th} iterations beginning with the 0th iteration (the initial estimates), where n is the value of the print interval. If the iteration history is requested, then the last iteration is always displayed regardless of n .
- **Lagrange multiplier test.** Displays Lagrange multiplier test statistics for assessing the validity of a scale parameter that is computed using the deviance or Pearson chi-square, or set at a fixed number, for the normal, gamma, and inverse Gaussian distributions. For the negative binomial distribution, this tests the fixed ancillary parameter.

Model Effects. The following options are available:

- **Analysis Type.** Specify the type of analysis to produce. Type I analysis is generally appropriate when you have a priori reasons for ordering predictors in the model, while Type III is more generally applicable. Wald or likelihood-ratio statistics are computed based upon the selection in the Chi-Square Statistics group.
- **Confidence Interval Level (%).** Specify a confidence level greater than 50 and less than 100. Wald intervals are based on the assumption that parameters have an asymptotic normal distribution; profile likelihood intervals are more accurate but can be computationally expensive. The tolerance level for profile likelihood intervals is the criteria used to stop the iterative algorithm used to compute the intervals.
- **Log-Likelihood Function.** This controls the display format of the log-likelihood function. The full function includes an additional term that is constant with respect to the parameter estimates; it has no effect on parameter estimation and is left out of the display in some software products.

Related information

- [GenLin Node](#)
 - [GenLin Node Field Options](#)
 - [GenLin Node Model Options](#)
 - [GenLin Node Expert Options](#)
 - [Generalized Linear Models Iterations](#)
-

GenLin Model Nugget

A GenLin model nugget represents the equations estimated by a GenLin node. They contain all of the information captured by the model, as well as information about the model structure and performance.

When you run a stream containing a GenLin model nugget, the node adds new fields whose contents depend on the nature of the target field:

- **Flag target.** Adds fields containing the predicted category and associated probability and the probabilities for each category. The names of the first two new fields are derived from the name of the output field being predicted, prefixed with $\$G-$ for the predicted category and $\$GP-$ for the associated probability. For example, for an output field named *default*, the new fields would be named $\$G-default$ and $\$GP-default$. The latter two additional fields are named based on the values of the output field, prefixed by $\$GP-$. For example, if the legal values of *default* are Yes and No, the new fields would be named $\$GP-Yes$ and $\$GP-No$.
- **Continuous target.** Adds fields containing the predicted mean and standard error.
- **Continuous target, representing number of events in a series of trials.** Adds fields containing the predicted mean and standard error.
- **Ordinal target.** Adds fields containing the predicted category and associated probability for each value of the ordered set. The names of the fields are derived from the value of the ordered set being predicted, prefixed with $\$G-$ for the predicted category and $\$GP-$ for the associated probability.

Generating a Filter node. The Generate menu allows you to create a new Filter node to pass input fields based on the results of the model.

Predictor Importance

Optionally, a chart that indicates the relative importance of each predictor in estimating the model may also be displayed on the Model tab. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least. Note this chart is only available if Calculate predictor importance is selected on the Analyze tab before generating the model. See the topic [Predictor Importance](#) for more information.

- [GenLin Model Nugget Advanced Output](#)
- [GenLin Model Nugget Settings](#)
- [GenLin Model Nugget Summary](#)

Related information

- [GenLin Model Nugget Advanced Output](#)
 - [GenLin Model Nugget Settings](#)
 - [GenLin Model Nugget Summary](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
 - [GenLin Node](#)
-

GenLin Model Nugget Advanced Output

The advanced output for generalized linear model gives detailed information about the estimated model and its performance. Most of the information contained in the advanced output is quite technical, and extensive knowledge of this type of analysis is required to properly interpret this output. See the topic [Generalized Linear Models Advanced Output](#) for more information.

Related information

- [GenLin Model Nugget](#)
 - [GenLin Model Nugget Settings](#)
 - [GenLin Model Nugget Summary](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

GenLin Model Nugget Settings

The Settings tab for a GenLin model nugget allows you to obtain propensity scores when scoring the model, and also for SQL generation during model scoring. This tab is available for models with flag targets only, and only after the model nugget has been added to a stream.

Calculate raw propensity scores. For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

Calculate adjusted propensity scores. Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Related information

- [GenLin Model Nugget](#)
- [GenLin Model Nugget Advanced Output](#)
- [GenLin Model Nugget Summary](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)

GenLin Model Nugget Summary

The Summary tab for a GenLin model nugget displays the fields and settings used to generate the model. In addition, if you have executed an Analysis node attached to this modeling node, information from that analysis will also be displayed in this section. For general information on using the model browser, see [Browsing model nuggets](#).

Related information

- [GenLin Model Nugget](#)
- [GenLin Model Nugget Advanced Output](#)
- [GenLin Model Nugget Settings](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)

Generalized Linear Mixed Models

- [GLMM Node](#)

GLMM Node

Use this node to create a generalized linear mixed model (GLMM).

- [Generalized linear mixed models](#)

Generalized linear mixed models

Generalized linear mixed models extend the linear model so that:

- The target is linearly related to the factors and covariates via a specified link function.
- The target can have a non-normal distribution.
- The observations can be correlated.

Generalized linear mixed models cover a wide variety of models, from simple linear regression to complex multilevel models for non-normal longitudinal data.

Examples. The district school board can use a generalized linear mixed model to determine whether an experimental teaching method is effective at improving math scores. Students from the same classroom should be correlated since they are taught by the same teacher, and classrooms within the same school may also be correlated, so we can include random effects at school and class levels to account for different sources of variability.

Medical researchers can use a generalized linear mixed model to determine whether a new anticonvulsant drug can reduce a patient's rate of epileptic seizures. Repeated measurements from the same patient are typically positively correlated so a mixed model with some random effects should be appropriate. The target field, the number of seizures, takes positive integer values, so a generalized linear mixed model with a Poisson distribution and log link may be appropriate.

Executives at a cable provider of television, phone, and internet services can use a generalized linear mixed model to know more about potential customers. Since possible answers have nominal measurement levels, the company analyst uses a generalized logit mixed model with a random intercept to capture correlation between answers to the service usage questions across service types (tv, phone, internet) within a given survey responder's answers.

The Data Structure tab allows you to specify the structural relationships between records in your dataset when observations are correlated. If the records in the dataset represent independent observations, you do not need to specify anything on this tab.

Subjects. The combination of values of the specified categorical fields should uniquely define subjects within the dataset. For example, a single *Patient ID* field should be sufficient to define subjects in a single hospital, but the combination of *Hospital ID* and *Patient ID* may be necessary if patient identification numbers are not unique across hospitals. In a repeated measures setting, multiple observations are recorded for each subject, so each subject may occupy multiple records in the dataset.

A **subject** is an observational unit that can be considered independent of other subjects. For example, the blood pressure readings from a patient in a medical study can be considered independent of the readings from other patients. Defining subjects becomes particularly important when there are repeated measurements per subject and you want to model the correlation between these observations. For example, you might expect that blood pressure readings from a single patient during consecutive visits to the doctor are correlated.

All of the fields specified as Subjects on the Data Structure tab are used to define subjects for the residual covariance structure, and provide the list of possible fields for defining subjects for random-effects covariance structures on the [Random Effect Block](#).

Repeated measures. The fields specified here are used to identify repeated observations. For example, a single variable *Week* might identify the 10 weeks of observations in a medical study, or *Month* and *Day* might be used together to identify daily observations over the course of a year.

Define covariance groups by. The categorical fields specified here define independent sets of repeated effects covariance parameters; one for each category defined by the cross-classification of the grouping fields. All subjects have the same covariance type; subjects within the same covariance grouping will have the same values for the parameters.

Spatial covariance coordinates. The variables in this list specify the coordinates of the repeated observations when one of the spatial covariance types is selected for the repeated covariance type.

Repeated covariance type. This specifies the covariance structure for the residuals. The available structures are:

- First-order autoregressive (AR1)
 - Autoregressive moving average (1,1) (ARMA11)
 - Compound symmetry
 - Diagonal
 - Scaled identity
 - Spatial: Power
 - Spatial: Exponential
 - Spatial: Gaussian
 - Spatial: Linear
 - Spatial: Linear-log
 - Spatial: Spherical
 - Toeplitz
 - Unstructured
 - Variance components
-
- [Target \(generalized linear mixed models\)](#)
 - [Fixed Effects \(generalized linear mixed models\)](#)
 - [Random Effects \(generalized linear mixed models\)](#)
 - [Weight and Offset \(generalized linear mixed models\)](#)
 - [General Build Options \(generalized linear mixed models\)](#)
 - [Estimation \(generalized linear mixed models\)](#)
 - [General \(generalized linear mixed models\)](#)
 - [Estimated Means \(generalized linear mixed models\)](#)
 - [Model view \(generalized linear mixed models\)](#)

Target (generalized linear mixed models)

These settings define the target, its distribution, and its relationship to the predictors through the link function.

Target. The target is required. It can have any measurement level, and the measurement level of the target restricts which distributions and link functions are appropriate.

- **Use number of trials as denominator.** When the target response is a number of events occurring in a set of trials, the target field contains the number of events and you can select an additional field containing the number of trials. For example, when testing a new pesticide you might expose samples of ants to different concentrations of the pesticide and then record the number of ants killed and the number of ants in each sample. In this case, the field recording the number of ants killed should be specified as the target (events) field, and the field recording the number of ants in each sample should be specified as the trials field. If the number of ants is the same for each sample, then the number of trials may be specified using a fixed value.
The number of trials should be greater than or equal to the number of events for each record. Events should be non-negative integers, and trials should be positive integers.
- **Customize reference category.** For a categorical target, you can choose the reference category. This can affect certain output, such as parameter estimates, but it should not change the model fit. For example, if your target takes values 0, 1, and 2, by default, the procedure makes the last (highest-valued) category, or 2, the reference category. In this situation, parameter estimates should be interpreted as relating to the likelihood of category 0 or 1 relative to the likelihood of category 2. If you specify a custom category and your target has defined labels, you can set the reference category by choosing a value from the list. This can be convenient when, in the middle of specifying a model, you don't remember exactly how a particular field was coded.

Target Distribution and Relationship (Link) with the Linear Model. Given the values of the predictors, the model expects the distribution of values of the target to follow the specified shape, and for the target values to be linearly related to the predictors through the specified link function. Short cuts for several common models are provided, or choose a Custom setting if there is a particular distribution and link function combination you want to fit that is not on the short list.

- **Linear model.** Specifies a normal distribution with an identity link, which is useful when the target can be predicted using a linear regression or ANOVA model.
- **Gamma regression.** Specifies a Gamma distribution with a log link, which should be used when the target contains all positive values and is skewed towards larger values.
- **Loglinear.** Specifies a Poisson distribution with a log link, which should be used when the target represents a count of occurrences in a fixed period of time.
- **Negative binomial regression.** Specifies a negative binomial distribution with a log link, which should be used when the target and denominator represent the number of trials required to observe k successes.
- **Multinomial logistic regression.** Specifies a multinomial distribution, which should be used when the target is a multi-category response. It uses either a cumulative logit link (ordinal outcomes) or a generalized logit link (multi-category nominal responses).
- **Binary logistic regression.** Specifies a binomial distribution with a logit link, which should be used when the target is a binary response predicted by a logistic regression model.
- **Binary probit.** Specifies a binomial distribution with a probit link, which should be used when the target is a binary response with an underlying normal distribution.
- **Interval censored survival.** Specifies a binomial distribution with a complementary log-log link, which is useful in survival analysis when some observations have no termination event.

Distribution

This selection specifies the distribution of the target. The ability to specify a non-normal distribution and non-identity link function is the essential improvement of the generalized linear mixed model over the linear mixed model. There are many possible distribution-link function combinations, and several may be appropriate for any given dataset, so your choice can be guided by a priori theoretical considerations or which combination seems to fit best.

Binomial

This distribution is appropriate only for a target that represents a binary response or number of events.

Gamma

This distribution is appropriate for a target with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.

Inverse Gaussian

This distribution is appropriate for a target with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.

Multinomial

This distribution is appropriate for a target that represents a multi-category response. The form of the model will depend on the measurement level of the target.

A **nominal** target will result in a nominal multinomial model in which a separate set of model parameters are estimated for each category of the target (except the reference category). The parameter estimates for a given predictor show the relationship between that predictor and the likelihood of each category of the target, relative to the reference category.

An **ordinal** target will result in an ordinal multinomial model in which the traditional intercept term is replaced with a set of **threshold** parameters that relate to the cumulative probability of the target categories.

Negative binomial

Negative binomial regression uses a negative binomial distribution with a log link, which should be used when the target represents a count of occurrences with high variance.

Normal

This is appropriate for a continuous target whose values take a symmetric, bell-shaped distribution about a central (mean) value.

Poisson

This distribution can be thought of as the number of occurrences of an event of interest in a fixed period of time and is appropriate for variables with non-negative integer values. If a data value is non-integer, less than 0, or missing, then the corresponding case is not used in the analysis.

Link function

The link function is a transformation of the target that allows estimation of the model. The following functions are available:

Identity

$f(x)=x$. The target is not transformed. This link can be used with any distribution, except the multinomial.

Complementary log-log

$f(x)=\log(-\log(1-x))$. This is appropriate only with the binomial or multinomial distribution.

Cauchit

$f(x) = \tan(\pi(x - 0.5))$. This is appropriate only with the binomial or multinomial distribution.

Log

$f(x)=\log(x)$. This link can be used with any distribution, except the multinomial.

Log complement

$f(x)=\log(1-x)$. This is appropriate only with the binomial distribution.

Logit

$f(x)=\log(x / (1-x))$. This is appropriate only with the binomial or multinomial distribution.

Negative log-log

$f(x)=-\log(-\log(x))$. This is appropriate only with the binomial or multinomial distribution.

Probit

$f(x)=\Phi^{-1}(x)$, where Φ^{-1} is the inverse standard normal cumulative distribution function. This is appropriate only with the binomial or multinomial distribution.

Power

$f(x)=x^\alpha$, if $\alpha \neq 0$. $f(x)=\log(x)$, if $\alpha=0$. α is the required number specification and must be a real number. This link can be used with any distribution, except the multinomial.

Related information

- [Generalized linear mixed models](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)

Fixed Effects (generalized linear mixed models)

Fixed effects factors are generally thought of as fields whose values of interest are all represented in the dataset, and can be used for scoring. By default, fields with the predefined input role that are not specified elsewhere in the dialog are entered in the fixed effects portion of the model. Categorical (flag, nominal, and ordinal) fields are used as factors in the model and continuous fields are used as covariates.

Enter effects into the model by selecting one or more fields in the source list and dragging to the effects list. The type of effect created depends upon which hotspot you drop the selection.

- **Main.** Dropped fields appear as separate main effects at the bottom of the effects list.
- **2-way.** All possible pairs of the dropped fields appear as 2-way interactions at the bottom of the effects list.
- **3-way.** All possible triplets of the dropped fields appear as 3-way interactions at the bottom of the effects list.
- *****. The combination of all dropped fields appear as a single interaction at the bottom of the effects list.

Buttons to the right of the Effect Builder allow you to perform various actions.

Table 1. Effect builder button descriptions

Icon	Description
	Delete terms from the fixed effects model by selecting the terms you want to delete and clicking the delete button.
	Reorder the terms within the fixed effects model by selecting the terms you want to reorder and clicking the up or down arrow.
	Add nested terms to the model using the Add a Custom Term (generalized linear mixed models) dialog, by clicking on the Add a Custom Term button.

Include Intercept. The intercept is usually included in the model. If you can assume the data pass through the origin, you can exclude the intercept.

- [Add a Custom Term \(generalized linear mixed models\)](#)

Related information

- [Generalized linear mixed models](#)
 - [Target \(generalized linear mixed models\)](#)
 - [Add a Custom Term \(generalized linear mixed models\)](#)
 - [Random Effects \(generalized linear mixed models\)](#)
 - [Random Effect Block \(generalized linear mixed models\)](#)
 - [Weight and Offset \(generalized linear mixed models\)](#)
 - [General Build Options \(generalized linear mixed models\)](#)
 - [Estimated Means \(generalized linear mixed models\)](#)
 - [Model view \(generalized linear mixed models\)](#)
 - [Model Summary \(generalized linear mixed models\)](#)
 - [Data Structure \(generalized linear mixed models\)](#)
 - [Predicted by Observed \(generalized linear mixed models\)](#)
 - [Classification \(generalized linear mixed models\)](#)
 - [Fixed Effects \(generalized linear mixed models\)](#)
 - [Fixed Coefficients \(generalized linear mixed models\)](#)
 - [Random Effect Covariances \(generalized linear mixed models\)](#)
 - [Covariance Parameters \(generalized linear mixed models\)](#)
 - [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
 - [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)
-

Add a Custom Term (generalized linear mixed models)

You can build nested terms for your model in this procedure. Nested terms are useful for modeling the effect of a factor or covariate whose values do not interact with the levels of another factor. For example, a grocery store chain may follow the spending habits of its customers at several store locations. Since each customer frequents only one of these locations, the *Customer* effect can be said to be **nested within** the *Store location* effect.

Additionally, you can include interaction effects, such as polynomial terms involving the same covariate, or add multiple levels of nesting to the nested term.

Limitations: Nested terms have the following restrictions:

- All factors within an interaction must be unique. Thus, if A is a factor, then specifying A*A is invalid.
- All factors within a nested effect must be unique. Thus, if A is a factor, then specifying A(A) is invalid.
- No effect can be nested within a covariate. Thus, if A is a factor and X is a covariate, then specifying A(X) is invalid.

Constructing a nested term

1. Select a factor or covariate that is nested within another factor, and then click the arrow button.
2. Click (Within).
3. Select the factor within which the previous factor or covariate is nested, and then click the arrow button.
4. Click Add Term.

Optionally, you can include interaction effects or add multiple levels of nesting to the nested term.

Related information

- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)

Random Effects (generalized linear mixed models)

Random effects factors are fields whose values in the data file can be considered a random sample from a larger population of values. They are useful for explaining excess variability in the target. By default, if you have selected more than one subject in the Data Structure tab, a Random Effect block will be created for each subject beyond the innermost subject. For example, if you selected School, Class, and Student as subjects on the Data Structure tab, the following random effect blocks are automatically created:

- Random Effect 1: subject is school (with no effects, intercept only)
- Random Effect 2: subject is school * class (no effects, intercept only)

You can work with random effects blocks in the following ways:

1. To add a new block, click Add Block... This opens the [Random Effect Block \(generalized linear mixed models\)](#) dialog.
 2. To edit an existing block, select the block you want to edit and click Edit Block... This opens the [Random Effect Block \(generalized linear mixed models\)](#) dialog.
 3. To delete one or more blocks, select the blocks you want to delete and click the delete button.
- [Random Effect Block \(generalized linear mixed models\)](#)

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)

Random Effect Block (generalized linear mixed models)

Enter effects into the model by selecting one or more fields in the source list and dragging to the effects list. The type of effect created depends upon which hotspot you drop the selection. Categorical (flag, nominal, and ordinal) fields are used as factors in the model and continuous fields are used as covariates.

- **Main.** Dropped fields appear as separate main effects at the bottom of the effects list.
- **2-way.** All possible pairs of the dropped fields appear as 2-way interactions at the bottom of the effects list.
- **3-way.** All possible triplets of the dropped fields appear as 3-way interactions at the bottom of the effects list.
- *****. The combination of all dropped fields appear as a single interaction at the bottom of the effects list.

Buttons to the right of the Effect Builder allow you to perform various actions.

Table 1. Effect builder button descriptions

Icon	Description
	Delete terms from the model by selecting the terms you want to delete and clicking the delete button.
	Reorder the terms within the model by selecting the terms you want to reorder and clicking the up or down arrow.
	Add nested terms to the model using the Add a Custom Term (generalized linear mixed models) dialog, by clicking on the Add a Custom Term button.

Include Intercept. The intercept is not included in the random effects model by default. If you can assume the data pass through the origin, you can exclude the intercept.

Display parameter predictions for this block. Specifies to display the random-effects parameter estimates.

Define covariance groups by. The categorical fields specified here define independent sets of random effects covariance parameters; one for each category defined by the cross-classification of the grouping fields. A different set of grouping fields can be specified for each random effect block. All subjects have the same covariance type; subjects within the same covariance grouping will have the same values for the parameters.

Subject combination. This allows you to specify random effect subjects from preset combinations of subjects from the Data Structure tab. For example, if *School*, *Class*, and *Student* are defined as subjects on the Data Structure tab, and in that order, then the Subject combination dropdown list will have None, *School*, *School * Class*, and *School * Class * Student* as options.

Random effect covariance type. This specifies the covariance structure for the residuals. The available structures are:

- First-order autoregressive (AR1)
- Heterogeneous autoregressive (ARH1)
- Autoregressive moving average (1,1) (ARMA11)
- Compound symmetry
- Heterogeneous compound symmetry (CSH)
- Diagonal
- Scaled identity
- Toeplitz
- Unstructured
- Variance components

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)

Weight and Offset (generalized linear mixed models)

Analysis weight. The scale parameter is an estimated model parameter related to the variance of the response. The analysis weights are "known" values that can vary from observation to observation. If the analysis weight field is specified, the scale parameter, which is related to the variance of the response, is divided by the analysis weight values for each observation. Records with analysis weight values that are less than or equal to 0 or are missing are not used in the analysis.

Offset. The offset term is a "structural" predictor. Its coefficient is not estimated by the model but is assumed to have the value 1; thus, the values of the offset are simply added to the linear predictor of the target. This is especially useful in Poisson regression models, where each case may have different levels of exposure to the event of interest.

For example, when modeling accident rates for individual drivers, there is an important difference between a driver who has been at fault in one accident in three years of experience and a driver who has been at fault in one accident in 25 years! The number of accidents can be modeled as a Poisson or negative binomial response with a log link if the natural log of the experience of the driver is included as an offset term.

Other combinations of distribution and link types would require other transformations of the offset variable.

Related information

- [Generalized linear mixed models](#)
 - [Target \(generalized linear mixed models\)](#)
 - [Fixed Effects \(generalized linear mixed models\)](#)
 - [Add a Custom Term \(generalized linear mixed models\)](#)
 - [Random Effects \(generalized linear mixed models\)](#)
 - [Random Effect Block \(generalized linear mixed models\)](#)
 - [General Build Options \(generalized linear mixed models\)](#)
 - [Estimated Means \(generalized linear mixed models\)](#)
 - [Model view \(generalized linear mixed models\)](#)
 - [Model Summary \(generalized linear mixed models\)](#)
 - [Data Structure \(generalized linear mixed models\)](#)
 - [Predicted by Observed \(generalized linear mixed models\)](#)
 - [Classification \(generalized linear mixed models\)](#)
 - [Fixed Effects \(generalized linear mixed models\)](#)
 - [Fixed Coefficients \(generalized linear mixed models\)](#)
 - [Random Effect Covariances \(generalized linear mixed models\)](#)
 - [Covariance Parameters \(generalized linear mixed models\)](#)
 - [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
 - [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)
-

General Build Options (generalized linear mixed models)

These selections specify some more advanced criteria used to build the model.

Sorting Order

These controls determine the order of the categories for the target and factors (categorical inputs) for purposes of determining the "last" category. The target sort order setting is ignored if the target is not categorical or if a custom reference category is specified on the [Target \(generalized linear mixed models\)](#) settings.

Stopping Rules

You can specify the maximum number of iterations the algorithm will execute. The algorithm uses a doubly iterative process that consists of an inner loop and an outer loop. The value that is specified for the maximum number of iterations applies to both loops. Specify a non-negative integer. The default is 100.

Post-Estimation Settings

These settings determine how some of the model output is computed for viewing.

Confidence level (%)

This is the level of confidence used to compute interval estimates of the model coefficients. Specify a value greater than 0 and less than 100. The default is 95.

Degrees of freedom

This specifies how degrees of freedom are computed for significance tests. Choose Residual method if your sample size is sufficiently large, or the data are balanced, or the model uses a simpler covariance type (for example, scaled identity or diagonal). This is the default setting. Choose Satterthwaite approximation if your sample size is small, or the data are unbalanced, or the model uses a complicated covariance type (for example, unstructured). Choose Kenward-Roger approximation if your sample size is small and you have a Restricted Maximum Likelihood (REML) model.

Tests of fixed effects and coefficients

This is the method for computing the parameter estimates covariance matrix. Choose the robust estimate if you are concerned that the model assumptions are violated.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)

- [Random Effect Covariances \(generalized linear mixed models\)](#)
 - [Covariance Parameters \(generalized linear mixed models\)](#)
 - [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
 - [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)
 - [Estimation \(generalized linear mixed models\)](#)
-

Estimation (generalized linear mixed models)

The model building algorithm uses a doubly iterative process that consists of an inner loop and an outer loop. The following settings apply to the inner loop.

Sorting Order

These controls determine the order of the categories for the target and factors (categorical inputs) for purposes of determining the "last" category. The target sort order setting is ignored if the target is not categorical or if a custom reference category is specified on the [Target \(generalized linear mixed models\)](#) settings.

Parameter Convergence.

Convergence is assumed if the maximum absolute change or maximum relative change in the parameter estimates is less than the value specified, which must be non-negative. The criterion is not used if the value specified equals 0.

Log-likelihood Convergence.

Convergence is assumed if the absolute change or relative change in the log-likelihood function is less than the value specified, which must be non-negative. The criterion is not used if the value specified equals 0.

Hessian Convergence.

For the Absolute specification, convergence is assumed if a statistic based on the Hessian is less than the value specified. For the Relative specification, convergence is assumed if the statistic is less than the product of the value specified and the absolute value of the log-likelihood. The criterion is not used if the value specified equals 0.

Maximum Fisher scoring steps.

Specify a non-negative integer. A value of 0 specifies the Newton-Raphson method. Values greater than 0 specify to use the Fisher scoring algorithm up to iteration number n , where n is the specified integer, and Newton-Raphson thereafter.

Singularity tolerance.

This value is used as the tolerance in checking singularity. Specify a positive value.

Stopping Rules

You can specify the maximum number of iterations the algorithm will execute. The algorithm uses a doubly iterative process that consists of an inner loop and an outer loop. The value that is specified for the maximum number of iterations applies to both loops. Specify a non-negative integer. The default is 100.

Post-Estimation Settings

These settings determine how some of the model output is computed for viewing.

Confidence level (%)

This is the level of confidence used to compute interval estimates of the model coefficients. Specify a value greater than 0 and less than 100. The default is 95.

Degrees of freedom

This specifies how degrees of freedom are computed for significance tests. Choose Residual method if your sample size is sufficiently large, or the data are balanced, or the model uses a simpler covariance type (for example, scaled identity or diagonal). This is the default setting. Choose Satterthwaite approximation if your sample size is small, or the data are unbalanced, or the model uses a complicated covariance type (for example, unstructured). Choose Kenward-Roger approximation if your sample size is small and you have a Restricted Maximum Likelihood (REML) model.

Tests of fixed effects and coefficients

This is the method for computing the parameter estimates covariance matrix. Choose the robust estimate if you are concerned that the model assumptions are violated.

Note: By default, Parameter Convergence is used, where the maximum Absolute change at a tolerance of 1E-6 is checked. This setting might produce results that differ from the results that are obtained in versions before version 22. To reproduce results from pre-22 versions, use Relative for the Parameter Convergence criterion and keep the default tolerance value of 1E-6.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)

- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)

General (generalized linear mixed models)

Model Name. You can generate the model name automatically based on the target fields or specify a custom name. The automatically generated name is the target field name. If there are multiple targets, then the model name is the field names in order, connected by ampersands. For example, if *field1* *field2* *field3* are targets, then the model name is: *field1 & field2 & field3*.

Make Available for Scoring. When the model is scored, the selected items in this group should be produced. The predicted value (for all targets) and confidence (for categorical targets) are always computed when the model is scored. The computed confidence can be based on the probability of the predicted value (the highest predicted probability) or the difference between the highest predicted probability and the second highest predicted probability.

- Predicted probability for categorical targets. This produces the predicted probabilities for categorical targets. A field is created for each category.
- Propensity scores for flag targets. For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. The model produces raw propensity scores; if partitions are in effect, the model also produces adjusted propensity scores based on the testing partition.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)
- [Settings \(generalized linear mixed models\)](#)

Estimated Means (generalized linear mixed models)

This tab allows you to display the estimated marginal means for levels of factors and factor interactions. Estimated marginal means are not available for multinomial models.

Terms

The model terms in the Fixed Effects that are entirely comprised of categorical fields are listed here. Check each term for which you want the model to produce estimated marginal means.

Contrast Type

This specifies the type of contrast to use for the levels of the contrast field.

None

No contrasts are produced.

Pairwise

Produces pairwise comparisons for all level combinations of the specified factors. This is the only available contrast for factor interactions.

Deviation

Contrasts compare each level of the factor to the grand mean.

Simple

Contrasts compare each level of the factor, except the last, to the last level. The "last" level is determined by the sort order for factors specified on the Build Options. Note that all of these contrast types are not orthogonal.

Contrast Field

This specifies a factor, the levels of which are compared using the selected contrast type. If None is selected as the contrast type, no contrast field can (or need) be selected.

Continuous Fields

The listed continuous fields are extracted from the terms in the Fixed Effects that use continuous fields. When computing estimated marginal means, covariates are fixed at the specified values. Select the mean or specify a custom value.

Adjust for multiple comparisons using

When performing hypothesis tests with multiple contrasts, the overall significance level can be adjusted from the significance levels for the included contrasts. This allows you to choose the adjustment method.

Least significant difference

This method does not control the overall probability of rejecting the hypotheses that some linear contrasts are different from the null hypothesis values.

Sequential Bonferroni

This is a sequentially step-down rejective Bonferroni procedure that is much less conservative in terms of rejecting individual hypotheses but maintains the same overall significance level.

Sequential Sidak

This is a sequentially step-down rejective Sidak procedure that is much less conservative in terms of rejecting individual hypotheses but maintains the same overall significance level.

The least significant difference method is less conservative than the sequential Sidak method, which in turn is less conservative than the sequential Bonferroni; that is, least significant difference will reject at least as many individual hypotheses as sequential Sidak, which in turn will reject at least as many individual hypotheses as sequential Bonferroni.

Display estimated means in terms of

This specifies whether to compute estimated marginal means based on the original scale of the target or based on the link function transformation.

Original target scale

Computes estimated marginal means for the target. Note that when the target is specified using the events/trials option, this gives the estimated marginal means for the events/trials proportion rather than for the number of events.

Link function transformation

Computes estimated marginal means for the linear predictor.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)

Model view (generalized linear mixed models)

By default, the Model Summary view is shown. To see another model view, select it from the view thumbnails.

For general information on Model objects, see [Models](#).

- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)
- [Settings \(generalized linear mixed models\)](#)

Model Summary (generalized linear mixed models)

This view is a snapshot, at-a-glance summary of the model and its fit.

Table. The table identifies the target, probability distribution, and link function specified on the [Target settings](#). If the target is defined by events and trials, the cell is split to show the events field and the trials field or fixed number of trials. Additionally the finite sample corrected Akaike information criterion (AICC) and Bayesian information criterion (BIC) are displayed.

- *Akaike Corrected*. A measure for selecting and comparing mixed models based on the -2 (Restricted) log likelihood. Smaller values indicate better models. The AICC "corrects" the AIC for small sample sizes. As the sample size increases, the AICC converges to the AIC.
- *Bayesian*. A measure for selecting and comparing models based on the -2 log likelihood. Smaller values indicate better models. The BIC also "penalizes" overparameterized models (complex models with a large number of inputs, for example), but more strictly than the AIC.

Chart. If the target is categorical, a chart displays the accuracy of the final model, which is the percentage of correct classifications.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)

Data Structure (generalized linear mixed models)

This view provides a summary of the data structure you specified, and helps you to check that the subjects and repeated measures have been specified correctly. The observed information for the first subject is displayed for each subject field and repeated measures field, and the target. Additionally, the number of levels for each subject field and repeated measures field is displayed.

Related information

- [Generalized linear mixed models](#)
 - [Target \(generalized linear mixed models\)](#)
 - [Fixed Effects \(generalized linear mixed models\)](#)
 - [Add a Custom Term \(generalized linear mixed models\)](#)
 - [Random Effects \(generalized linear mixed models\)](#)
 - [Random Effect Block \(generalized linear mixed models\)](#)
 - [Weight and Offset \(generalized linear mixed models\)](#)
 - [General Build Options \(generalized linear mixed models\)](#)
 - [Estimated Means \(generalized linear mixed models\)](#)
 - [Model view \(generalized linear mixed models\)](#)
 - [Model Summary \(generalized linear mixed models\)](#)
 - [Predicted by Observed \(generalized linear mixed models\)](#)
 - [Classification \(generalized linear mixed models\)](#)
 - [Fixed Effects \(generalized linear mixed models\)](#)
 - [Fixed Coefficients \(generalized linear mixed models\)](#)
 - [Random Effect Covariances \(generalized linear mixed models\)](#)
 - [Covariance Parameters \(generalized linear mixed models\)](#)
 - [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
 - [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)
-

Predicted by Observed (generalized linear mixed models)

For continuous targets, including targets specified as events/trials, this displays a binned scatterplot of the predicted values on the vertical axis by the observed values on the horizontal axis. Ideally, the points should lie on a 45-degree line; this view can tell you whether any records are predicted particularly badly by the model.

Related information

- [Generalized linear mixed models](#)
 - [Target \(generalized linear mixed models\)](#)
 - [Fixed Effects \(generalized linear mixed models\)](#)
 - [Add a Custom Term \(generalized linear mixed models\)](#)
 - [Random Effects \(generalized linear mixed models\)](#)
 - [Random Effect Block \(generalized linear mixed models\)](#)
 - [Weight and Offset \(generalized linear mixed models\)](#)
 - [General Build Options \(generalized linear mixed models\)](#)
 - [Estimated Means \(generalized linear mixed models\)](#)
 - [Model view \(generalized linear mixed models\)](#)
 - [Model Summary \(generalized linear mixed models\)](#)
 - [Data Structure \(generalized linear mixed models\)](#)
 - [Classification \(generalized linear mixed models\)](#)
 - [Fixed Effects \(generalized linear mixed models\)](#)
 - [Fixed Coefficients \(generalized linear mixed models\)](#)
 - [Random Effect Covariances \(generalized linear mixed models\)](#)
 - [Covariance Parameters \(generalized linear mixed models\)](#)
 - [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
 - [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)
-

Classification (generalized linear mixed models)

For categorical targets, this displays the cross-classification of observed versus predicted values in a heat map, plus the overall percent correct.

Table styles. There are several different display styles, which are accessible from the Style dropdown list.

- **Row percents.** This displays the row percentages (the cell counts expressed as a percent of the row totals) in the cells. This is the default.
- **Cell counts.** This displays the cell counts in the cells. The shading for the heat map is still based on the row percentages.
- **Heat map.** This displays no values in the cells, just the shading.
- **Compressed.** This displays no row or column headings, or values in the cells. It can be useful when the target has a lot of categories.

Missing. If any records have missing values on the target, they are displayed in a (Missing) row under all valid rows. Records with missing values do not contribute to the overall percent correct.

Multiple targets. If there are multiple categorical targets, then each target is displayed in a separate table and there is a Target dropdown list that controls which target to display.

Large tables. If the displayed target has more than 100 categories, no table is displayed.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)

Fixed Effects (generalized linear mixed models)

This view displays the size of each fixed effect in the model.

Styles. There are different display styles, which are accessible from the Style dropdown list.

- **Diagram.** This is a chart in which effects are sorted from top to bottom in the order in which they were specified on the Fixed Effects settings. Connecting lines in the diagram are weighted based on effect significance, with greater line width corresponding to more significant effects (smaller p -values). This is the default.
- **Table.** This is an ANOVA table for the overall model and the individual model effects. The individual effects are sorted from top to bottom in the order in which they were specified on the Fixed Effects settings.

Significance. There is a Significance slider that controls which effects are shown in the view. Effects with significance values greater than the slider value are hidden. This does not change the model, but simply allows you to focus on the most important effects. By default the value is 1.00, so that no effects are filtered based on significance.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)

- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)
-

Fixed Coefficients (generalized linear mixed models)

This view displays the value of each fixed coefficient in the model. Note that factors (categorical predictors) are indicator-coded within the model, so that **effects** containing factors will generally have multiple associated **coefficients**; one for each category except the category corresponding to the redundant coefficient.

Styles. There are different display styles, which are accessible from the Style dropdown list.

- **Diagram.** This is a chart which displays the intercept first, and then sorts effects from top to bottom in the order in which they were specified on the Fixed Effects settings. Within effects containing factors, coefficients are sorted by ascending order of data values. Connecting lines in the diagram are colored and weighted based on coefficient significance, with greater line width corresponding to more significant coefficients (smaller p -values). This is the default style.
- **Table.** This shows the values, significance tests, and confidence intervals for the individual model coefficients. After the intercept, the effects are sorted from top to bottom in the order in which they were specified on the Fixed Effects settings. Within effects containing factors, coefficients are sorted by ascending order of data values.

Multinomial. If the multinomial distribution is in effect, then the Multinomial drop-down list controls which target category to display. The sort order of the values in the list is determined by the specification on the Build Options settings.

Exponential. This displays exponential coefficient estimates and confidence intervals for certain model types, including Binary logistic regression (binomial distribution and logit link), Nominal logistic regression (multinomial distribution and logit link), Negative binomial regression (negative binomial distribution and log link), and Log-linear model (Poisson distribution and log link).

Significance. There is a Significance slider that controls which coefficients are shown in the view. Coefficients with significance values greater than the slider value are hidden. This does not change the model, but simply allows you to focus on the most important coefficients. By default the value is 1.00, so that no coefficients are filtered based on significance.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)

Random Effect Covariances (generalized linear mixed models)

This view displays the random effects covariance matrix (\mathbf{G}).

Styles. There are different display styles, which are accessible from the Style dropdown list.

- **Covariance values.** This is a heat map of the covariance matrix in which effects are sorted from top to bottom in the order in which they were specified on the Fixed Effects settings. Colors in the corrgram correspond to the cell values as shown in the key. This is the default.
- **Corrgram.** This is a heat map of the covariance matrix.
- **Compressed.** This is a heat map of the covariance matrix without the row and column headings.

Blocks. If there are multiple random effect blocks, then there is a Block dropdown list for selecting the block to display.

Groups. If a random effect block has a group specification, then there is a Group dropdown list for selecting the group level to display.

Multinomial. If the multinomial distribution is in effect, then the Multinomial drop-down list controls which target category to display. The sort order of the values in the list is determined by the specification on the Build Options settings.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)

Covariance Parameters (generalized linear mixed models)

This view displays the covariance parameter estimates and related statistics for residual and random effects. These are advanced, but fundamental, results that provide information on whether the covariance structure is suitable.

Summary table. This is a quick reference for the number of parameters in the residual (**R**) and random effect (**G**) covariance matrices, the rank (number of columns) in the fixed effect (**X**) and random effect (**Z**) design matrices, and the number of subjects defined by the subject fields that define the data structure.

Covariance parameter table. For the selected effect, the estimate, standard error, and confidence interval are displayed for each covariance parameter. The number of parameters shown depends upon the covariance structure for the effect and, for random effect blocks, the number of effects in the block. If you see that the off-diagonal parameters are not significant, you may be able to use a simpler covariance structure.

Effects. If there are random effect blocks, then there is an Effect dropdown list for selecting the residual or random effect block to display. The residual effect is always available.

Groups. If a residual or random effect block has a group specification, then there is a Group dropdown list for selecting the group level to display.

Multinomial. If the multinomial distribution is in effect, then the Multinomial drop-down list controls which target category to display. The sort order of the values in the list is determined by the specification on the Build Options settings.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)

- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)
-

Estimated Means: Significant Effects (generalized linear mixed models)

These are charts displayed for the 10 "most significant" fixed all-factor effects, starting with the three-way interactions, then the two-way interactions, and finally main effects. The chart displays the model-estimated value of the target on the vertical axis for each value of the main effect (or first listed effect in an interaction) on the horizontal axis; a separate line is produced for each value of the second listed effect in an interaction; a separate chart is produced for each value of the third listed effect in a three-way interaction; all other predictors are held constant. It provides a useful visualization of the effects of each predictor's coefficients on the target. Note that if no predictors are significant, no estimated means are produced.

Confidence. This displays upper and lower confidence limits for the marginal means, using the confidence level specified as part of the Build Options.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)

Estimated Means: Custom Effects (generalized linear mixed models)

These are tables and charts for user-requested fixed all-factor effects.

Styles. There are different display styles, which are accessible from the Style dropdown list.

- **Diagram.** This style displays a line chart of the model-estimated value of the target on the vertical axis for each value of the main effect (or first listed effect in an interaction) on the horizontal axis; a separate line is produced for each value of the second listed effect in an interaction; a separate chart is produced for each value of the third listed effect in a three-way interaction; all other predictors are held constant.

If contrasts were requested, another chart is displayed to compare levels of the contrast field; for interactions, a chart is displayed for each level combination of the effects other than the contrast field. For **pairwise** contrasts, it is a distance network chart; that is, a graphical representation of the comparisons table in which the distances between nodes in the network correspond to differences between samples. Yellow lines correspond to statistically significant differences; black lines correspond to non-significant differences. Hovering over a line in the network displays a tooltip with the adjusted significance of the difference between the nodes connected by the line.

For **deviation** contrasts, a bar chart is displayed with the model-estimated value of the target on the vertical axis and the values of the contrast field on the horizontal axis; for interactions, a chart is displayed for each level combination of the effects other than the contrast field. The bars show the difference between each level of the contrast field and the overall mean, which is represented by a black horizontal line.

For **simple** contrasts, a bar chart is displayed with the model-estimated value of the target on the vertical axis and the values of the contrast field on the horizontal axis; for interactions, a chart is displayed for each level combination of the effects other than the contrast field. The bars show the difference between each level of the contrast field (except the last) and the last level, which is represented by a black horizontal line.

- **Table.** This style displays a table of the model-estimated value of the target, its standard error, and confidence interval for each level combination of the fields in the effect; all other predictors are held constant. If contrasts were requested, another table is displayed with the estimate, standard error, significance test, and confidence interval for each contrast; for interactions, there a separate set of rows for each level combination of the effects other than the contrast field. Additionally, a table with the overall test results is displayed; for interactions, there is a separate overall test for each level combination of the effects other than the contrast field.

Confidence. This toggles the display of upper and lower confidence limits for the marginal means, using the confidence level specified as part of the Build Options.

Layout. This toggles the layout of the pairwise contrasts diagram. The circle layout is less revealing of contrasts than the network layout but avoids overlapping lines.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)
- [General Build Options \(generalized linear mixed models\)](#)
- [Estimated Means \(generalized linear mixed models\)](#)
- [Model view \(generalized linear mixed models\)](#)
- [Model Summary \(generalized linear mixed models\)](#)
- [Data Structure \(generalized linear mixed models\)](#)
- [Predicted by Observed \(generalized linear mixed models\)](#)
- [Classification \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Fixed Coefficients \(generalized linear mixed models\)](#)
- [Random Effect Covariances \(generalized linear mixed models\)](#)
- [Covariance Parameters \(generalized linear mixed models\)](#)
- [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)

Settings (generalized linear mixed models)

When the model is scored, the selected items in this tab should be produced. The predicted value (for all targets) and confidence (for categorical targets) are always computed when the model is scored. The computed confidence can be based on the probability of the predicted value (the highest predicted probability) or the difference between the highest predicted probability and the second highest predicted probability.

- Predicted probability for categorical targets. This produces the predicted probabilities for categorical targets. A field is created for each category.
- Propensity scores for flag targets. For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. The model produces raw propensity scores; if partitions are in effect, the model also produces adjusted propensity scores based on the testing partition.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Related information

- [Generalized linear mixed models](#)
- [Target \(generalized linear mixed models\)](#)
- [Fixed Effects \(generalized linear mixed models\)](#)
- [Add a Custom Term \(generalized linear mixed models\)](#)
- [Random Effects \(generalized linear mixed models\)](#)
- [Random Effect Block \(generalized linear mixed models\)](#)
- [Weight and Offset \(generalized linear mixed models\)](#)

- [General Build Options \(generalized linear mixed models\)](#)
 - [General \(generalized linear mixed models\)](#)
 - [Estimated Means \(generalized linear mixed models\)](#)
 - [Model view \(generalized linear mixed models\)](#)
 - [Model Summary \(generalized linear mixed models\)](#)
 - [Data Structure \(generalized linear mixed models\)](#)
 - [Predicted by Observed \(generalized linear mixed models\)](#)
 - [Classification \(generalized linear mixed models\)](#)
 - [Fixed Effects \(generalized linear mixed models\)](#)
 - [Fixed Coefficients \(generalized linear mixed models\)](#)
 - [Random Effect Covariances \(generalized linear mixed models\)](#)
 - [Covariance Parameters \(generalized linear mixed models\)](#)
 - [Estimated Means: Significant Effects \(generalized linear mixed models\)](#)
 - [Estimated Means: Custom Effects \(generalized linear mixed models\)](#)
-

GLE Node

The GLE model identifies the dependent variable that is linearly related to the factors and covariates via a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers widely used statistical models, such as linear regression for normally distributed responses, logistic models for binary data, loglinear models for count data, complementary log-log models for interval-censored survival data, plus many other statistical models through its very general model formulation.

Examples. A shipping company can use generalized linear models to fit a Poisson regression to damage counts for several types of ships constructed in different time periods, and the resulting model can help determine which ship types are most prone to damage.

A car insurance company can use generalized linear models to fit a gamma regression to damage claims for cars, and the resulting model can help determine the factors that contribute the most to claim size.

Medical researchers can use generalized linear models to fit a complementary log-log regression to interval-censored survival data to predict the time to recurrence for a medical condition.

GLE models work by building an equation that relates the input field values to the output field values. Once the model is generated, it can be used to estimate values for new data.

For a categorical target, for each record, a probability of membership is computed for each possible output category. The target category with the highest probability is assigned as the predicted output value for that record.

Requirements. You need one or more input fields and exactly one target field (which can have a measurement level of *Continuous*, *Categorical*, or *Flag*) with two or more categories. Fields used in the model must have their types fully instantiated.

- [Target \(GLE models\)](#)
 - [Model effects \(GLE models\)](#)
 - [Weight and Offset \(GLE models\)](#)
 - [Build options \(GLE models\)](#)
 - [Estimation \(GLE models\)](#)
 - [Model selection \(GLE models\)](#)
 - [Model options \(GLE models\)](#)
 - [GLE model nugget](#)
-

Target (GLE models)

These settings define the target, its distribution, and its relationship to the predictors through the link function.

Target The target is required. It can have any measurement level, and the measurement level of the target affects which distributions and link functions are appropriate.

- Use predefined target To use the target settings from an upstream Type node (or the Types tab of an upstream source node), select this option.
- Use custom target To manually assign a target, select this option.
- Use number of trials as denominator When the target response is a number of events occurring in a set of trials, the target field contains the number of events and you can select an additional field containing the number of trials. For example, when testing a new pesticide you might expose samples of ants to different concentrations of the pesticide and then record the number of ants killed and the number of ants in each sample. In this case, the field recording the number of ants killed should be specified as the target (events) field, and the field recording the number of ants in each sample should be specified as the trials field. If the number of ants is the same for each sample, then the number of trials may be specified using a fixed value.

The number of trials should be greater than or equal to the number of events for each record. Events should be non-negative integers, and trials should be positive integers.

- Customize reference category. For a categorical target, you can choose the reference category. This can affect certain output, such as parameter estimates, but it should not change the model fit. For example, if your target takes values 0, 1, and 2, by default, the procedure makes the last (highest-valued) category, or 2, the reference category. In this situation, parameter estimates should be interpreted as relating to the likelihood of category 0 or 1 *relative* to the likelihood of category 2. If you specify a custom category and your target has defined labels, you can set the reference category by choosing a value from the list. This can be convenient when, in the middle of specifying a model, you don't remember exactly how a particular field was coded.

Target Distribution and Relationship (Link) with the Linear Model Given the values of the predictors, the model expects the distribution of values of the target to follow the specified shape, and for the target values to be linearly related to the predictors through the specified link function. Short cuts for several common models are provided, or choose a Custom setting if there is a particular distribution and link function combination you want to fit that is not on the short list.

- Linear model Specifies a normal distribution with an identity link, which is useful when the target can be predicted using a linear regression or ANOVA model.
- Gamma regression Specifies a Gamma distribution with a log link, which should be used when the target contains all positive values and is skewed towards larger values.
- Loglinear Specifies a Poisson distribution with a log link, which should be used when the target represents a count of occurrences in a fixed period of time.
- Negative binomial regression Specifies a negative binomial distribution with a log link, which should be used when the target and denominator represent the number of trials required to observe k successes.
- Tweedie regression Specifies a Tweedie distribution with identity, log, or power link functions and are useful for modeling responses that are a mixture of zeros and positive real values. These distributions are also called *compound Poisson*, *compound gamma*, and *Poisson-gamma* distributions.
- Multinomial logistic regression Specifies a multinomial distribution, which should be used when the target is a multi-category response. It uses either a cumulative logit link (ordinal outcomes) or a generalized logit link (multi-category nominal responses).
- Binary logistic regression Specifies a binomial distribution with a logit link, which should be used when the target is a binary response predicted by a logistic regression model.
- Binary probit Specifies a binomial distribution with a probit link, which should be used when the target is a binary response with an underlying normal distribution.
- Interval censored survival Specifies a binomial distribution with a complementary log-log link, which is useful in survival analysis when some observations have no termination event.
- Custom Specify your own combination of distribution and link function.

Distribution

This selection specifies the Distribution of the target. The ability to specify a non-normal distribution and non-identity link function is the essential improvement of the generalized linear model over the linear model. There are many possible distribution-link function combinations, and several may be appropriate for any given dataset, so your choice can be guided by a priori theoretical considerations or which combination seems to fit best.

- Automatic If you are unsure which distribution to use, select this option; the node analyzes your data to estimate and apply the best distribution method.
- Binomial This distribution is appropriate only for a target that represents a binary response or number of events.
- Gamma This distribution is appropriate for a target with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.
- Inverse Gaussian This distribution is appropriate for a target with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.
- Multinomial This distribution is appropriate for a target that represents a multi-category response. The form of the model will depend on the measurement level of the target.

A **nominal** target will result in a nominal multinomial model in which a separate set of model parameters are estimated for each category of the target (except the reference category). The parameter estimates for a given predictor show the relationship between that predictor and the likelihood of each category of the target, relative to the reference category.

An **ordinal** target will result in an ordinal multinomial model in which the traditional intercept term is replaced with a set of **threshold** parameters that relate to the cumulative probability of the target categories.

- Negative binomial Negative binomial regression uses a negative binomial distribution with a log link, which should be used when the target represents a count of occurrences with high variance.
- Normal This is appropriate for a continuous target whose values take a symmetric, bell-shaped distribution about a central (mean) value.
- Poisson This distribution can be thought of as the number of occurrences of an event of interest in a fixed period of time and is appropriate for variables with non-negative integer values. If a data value is non-integer, less than 0, or missing, then the corresponding case is not used in the analysis.
- Tweedie This distribution is appropriate for variables that can be represented by Poisson mixtures of gamma distributions; the distribution is "mixed" in the sense that it combines properties of continuous (takes non-negative real values) and discrete distributions (positive probability mass at a single value, 0). The dependent variable must be numeric, with data values greater than or equal to zero. If a data

value is less than zero or missing, then the corresponding case is not used in the analysis. The fixed value of the Tweedie distribution's parameter can be any number greater than one and less than two.

Link functions

The Link function is a transformation of the target that allows estimation of the model. The following functions are available:

- Automatic If you are unsure which link to use, select this option; the node analyzes your data to estimate and apply the best link function.
- Identity $f(x)=x$. The target is not transformed. This link can be used with any distribution, except the multinomial.
- Complementary log-log $f(x)=\log(-\log(1-x))$. This is appropriate only with the binomial or multinomial distribution.
- Cauchit $f(x) = \tan(\pi(x - 0.5))$. This is appropriate only with the binomial or multinomial distribution.
- Log $f(x)=\log(x)$. This link can be used with any distribution, except the multinomial.
- Log complement $f(x)=\log(1-x)$. This is appropriate only with the binomial distribution.
- Logit $f(x)=\log(x / (1-x))$. This is appropriate only with the binomial or multinomial distribution.
- Negative log-log $f(x)=-\log(-\log(x))$. This is appropriate only with the binomial or multinomial distribution.
- Probit $f(x)=\Phi^{-1}(x)$, where Φ^{-1} is the inverse standard normal cumulative distribution function. This is appropriate only with the binomial or multinomial distribution.
- Power $f(x)=x^\alpha$, if $\alpha \neq 0$. $f(x)=\log(x)$, if $\alpha=0$. α is the required number specification and must be a real number. This link can be used with any distribution, except the multinomial.

Parameter for Tweedie Only available if you have selected either the Tweedie regression radio button or Tweedie as the Distribution method. Select a value between 1 and 2.

Model effects (GLE models)

Fixed effects factors are generally thought of as fields whose values of interest are all represented in the dataset, and can be used for scoring. By default, fields with the predefined input role that are not specified elsewhere in the dialog are entered in the fixed effects portion of the model. Categorical (flag, nominal, and ordinal) fields are used as factors in the model and continuous fields are used as covariates.

Enter effects into the model by selecting one or more fields in the source list and dragging to the effects list. The type of effect created depends upon which hotspot you drop the selection.

- Main Dropped fields appear as separate main effects at the bottom of the effects list.
- 2-way All possible pairs of the dropped fields appear as 2-way interactions at the bottom of the effects list.
- 3-way All possible triplets of the dropped fields appear as 3-way interactions at the bottom of the effects list.
- * The combination of all dropped fields appear as a single interaction at the bottom of the effects list.

Buttons to the right of the Effect Builder allow you to perform various actions.

Table 1. Effect builder button descriptions

Icon	Description
	Delete terms from the fixed effects model by selecting the terms you want to delete and clicking the delete button.
	Reorder the terms within the fixed effects model by selecting the terms you want to reorder and clicking the up or down arrow.
	Add nested terms to the model using the Add a Custom Term dialog, by clicking on the Add a Custom Term button.

Include Intercept The intercept is usually included in the model. If you can assume the data pass through the origin, you can exclude the intercept.

- [Add a custom term \(GLE models\)](#)

Add a custom term (GLE models)

You can build nested terms for your model in this procedure. Nested terms are useful for modeling the effect of a factor or covariate whose values do not interact with the levels of another factor. For example, a grocery store chain may follow the spending habits of its customers at several store locations. Since each customer frequents only one of these locations, the Customer effect can be said to be *nested within* the Store location effect.

Additionally, you can include interaction effects, such as polynomial terms involving the same covariate, or add multiple levels of nesting to the nested term.

Limitations. Nested terms have the following restrictions:

- All factors within an interaction must be unique. Thus, if A is a factor, then specifying A*A is invalid.
- All factors within a nested effect must be unique. Thus, if A is a factor, then specifying A(A) is invalid.
- No effect can be nested within a covariate. Thus, if A is a factor and X is a covariate, then specifying A(X) is invalid.

Constructing a nested term

1. Select a factor or covariate that is nested within another factor, and then click the arrow button.
2. Click (Within).
3. Select the factor within which the previous factor or covariate is nested, and then click the arrow button.
4. Click Add Term.

Optionally, you can include interaction effects or add multiple levels of nesting to the nested term.

Weight and Offset (GLE models)

Analysis weight The scale parameter is an estimated model parameter related to the variance of the response. The analysis weights are "known" values that can vary from observation to observation. If the Analysis weight field is specified, the scale parameter, which is related to the variance of the response, is divided by the analysis weight values for each observation. Records with analysis weight values that are less than or equal to 0, or are missing, are not used in the analysis.

Offset The offset term is a *structural* predictor. Its coefficient is not estimated by the model but is assumed to have the value 1; thus, the values of the offset are simply added to the linear predictor of the target. This is especially useful in Poisson regression models, where each case may have different levels of exposure to the event of interest.

For example, when modeling accident rates for individual drivers, there is an important difference between a driver who has been at fault in one accident in three years of experience and a driver who has been at fault in one accident in 25 years. The number of accidents can be modeled as a Poisson or negative binomial response with a log link if the natural log of the experience of the driver is included as an offset term.

Other combinations of distribution and link types would require other transformations of the offset variable.

Build options (GLE models)

These selections specify some more advanced criteria used to build the model.

Sorting Order These controls determine the order of the categories for the target and factors (categorical inputs) for purposes of determining the "last" category. The target sort order setting is ignored if the target is not categorical or if a custom reference category is specified on the [Target \(GLE models\)](#) settings.

Post-Estimation Settings These settings determine how some of the model output is computed for viewing.

- **Confidence level %** This is the level of confidence used to compute interval estimates of the model coefficients. Specify a value greater than 0 and less than 100. The default is 95.
- **Degrees of freedom** This specifies how degrees of freedom are computed for significance tests. Choose Fixed for all tests (Residual method) if your sample size is sufficiently large, or the data are balanced, or the model uses a simpler covariance type; for example, scaled identity or diagonal. This is the default. Choose Varied across tests (Satterthwaite approximation) if your sample size is small, or the data are unbalanced, or the model uses a complicated covariance type; for example, unstructured.
- **Tests of fixed effects and coefficients.** This is the method for computing the parameter estimates covariance matrix. Choose the robust estimate if you are concerned that the model assumptions are violated.

Detect influential outliers For all distributions except multinomial distribution, select this option to identify influential outliers.

Conduct trend analysis For a scatter plot, select this option to conduct trend analysis.

Estimation (GLE models)

Method Select the maximum likelihood estimation method to be used; the available options are:

- Fisher scoring
- Newton-Raphson
- Hybrid

Maximum Fisher iterations Specify a non-negative integer. A value of 0 specifies the Newton-Raphson method. Values greater than 0 specify to use the Fisher scoring algorithm up to iteration number n , where n is the specified integer, and Newton-Raphson thereafter.

Scale parameter method Select the method for the estimation of the scale parameter; the available options are:

- Maximum likelihood estimate
- Fixed value. You also set the Value to be used.
- Deviance
- Pearson Chi-square

Negative Binomial method Select the method for the estimation of the negative binomial ancillary parameter; the available options are:

- Maximum likelihood estimate
- Fixed value. You also set the Value to be used.

Perform non negative least squares. Select this option to perform non-negative least squares (NNLS) estimation. NNLS is a type of constrained least squares problem where the coefficients are not allowed to become negative. Not all data sets are suitable for NNLS, which requires a positive or no correlation between predictors and target.

Parameter Convergence Convergence is assumed if the maximum absolute change or maximum relative change in the parameter estimates is less than the value specified, which must be non-negative. The criterion is not used if the value specified equals 0.

Log-Likelihood Convergence Convergence is assumed if the absolute change or relative change in the log-likelihood function is less than the value specified, which must be non-negative. The criterion is not used if the value specified equals 0.

Hessian Convergence For the Absolute specification, convergence is assumed if a statistic based on the Hessian is less than the value specified. For the Relative specification, convergence is assumed if the statistic is less than the product of the value specified and the absolute value of the log-likelihood. The criterion is not used if the value specified equals 0.

Maximum iterations You can specify the maximum number of iterations the algorithm will execute. The algorithm uses a doubly iterative process that consists of an inner loop and an outer loop. The value that is specified for the maximum number of iterations applies to both loops. Specify a non-negative integer. The default is 100.

Singularity tolerance This value is used as the tolerance in checking singularity. Specify a positive value.

Note: By default, Parameter Convergence is used, where the maximum Absolute change at a tolerance of 1E-6 is checked. This setting might produce results that differ from the results that are obtained in versions before version 17. To reproduce results from pre-17 versions, use Relative for the Parameter Convergence criterion and keep the default tolerance value of 1E-6.

Model selection (GLE models)

Use model selection or regularization. To activate the controls on this pane, select this check box.

Method. Select the method of model selection or (if using Ridge) the regularization to be used. You can choose from the following options:

- Lasso. Also known as L1 Regularization, this method is faster than Forward Stepwise if there are a large number of predictors. This method prevents overfitting by shrinking (that is, by imposing a penalty) on the parameters. It can shrink some parameters to zero, performing a variable selection lasso.
- Ridge. Also known as L2 Regularization, this method prevents overfitting by shrinking (that is, by imposing a penalty) on the parameters. It shrinks all the parameters by the same proportions but eliminates none and is not a variable selection method.
- Elastic Net. Also known as L1 + L2 Regularization, this method prevents overfitting by shrinking (that is, by imposing a penalty) on the parameters. It can shrink some parameters to zero, performing variable selection.
- Forward Stepwise. This method starts with no effects in the model and adds or removes effects one step at a time until no more can be added or removed, according to the stepwise criteria.

Automatically detect two-way interactions. To automatically detect two-way interactions, select this option.

Penalty Parameters

These options are only available if you select either the Lasso or Elastic Net Method.

Automatically select penalty parameters. If you are unsure what parameter penalties to set, select this check box and the node identifies and applies the penalties.

Lasso penalty parameter. Enter the penalty parameter to be used by the Lasso model selection Method.

Elastic net penalty parameter 1. Enter the L1 penalty parameter to be used by the Elastic Net model selection Method.

Elastic net penalty parameter 2. Enter the L2 penalty parameter to be used by the Elastic Net model selection Method.

Forward Stepwise

These options are only available if you select the Forward Stepwise Method.

Include effects with p-value no less than. Specify the minimum probability value that effects can have to be included in the calculation.

Remove effects with p-value greater than. Specify the maximum probability value that effects can have to be included in the calculation.

Customize maximum number of effects in the final model. To activate the Maximum number of effects option, select this check box.

Maximum number of effects. Specify the maximum number of effects when using the forward stepwise building method. To optimize performance, 10 is the highest supported number.

Customize maximum number of steps. To activate the Maximum number of steps option, select this check box.

Maximum number of steps. Specify the maximum number of steps when using the forward stepwise building method.

Model options (GLE models)

Model name You can generate the model name automatically based on the target field, or specify a Custom name. The automatically generated name is the target field name. If there are multiple targets, then the model name is the field names in order, connected by ampersands. For example, if field1, field2, and field3 are targets, then the model name is: *field1 & field2 & field3*.

Calculate predictor importance For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may take longer to calculate for some models, particularly when working with large datasets, and is off by default for some models as a result.

For more information, see [Predictor Importance](#).

GLE model nugget

- [GLE model nugget output](#)
- [GLE model nugget settings](#)

GLE model nugget output

After you create a GLE model, the following information is available in the output.

Model Information table

The Model Information table provides key information about the model. The table identifies some high-level model settings, such as:

- The name of the target field selected in either the Type node or the GLE node Fields tab.
- The modeled and reference target category percentages.
- The probability distribution and associated link function.
- The model building method used.
- The number of predictors input and the number in the final model.
- The classification accuracy percentage.
- The model type.
- The percentage accuracy of the model, if the target is continuous.

Records Summary

The summary table shows how many records were used to fit the model, and how many were excluded. The details shown include the number and percentage of the records included and excluded, as well as the unweighted number if you used frequency weighting.

Predictor Importance

The Predictor Importance graph shows the importance of the top 10 inputs (predictors) in the model as a bar chart.

If there are more than 10 fields in the chart, you can change the selection of predictors that are included in the chart by using the slider beneath the chart. The indicator marks on the slider are a fixed width, and each mark on the slider represents 10 fields. You can move the indicator marks along the slider to display the next or previous 10 fields, ordered by predictor importance.

You can double-click the chart to open a separate dialog box in which you can edit the graph settings. For example, you can amend items such as the size of the graph, and the size and color of the fonts used. When you close this separate editing dialog box, the changes are applied to the chart that is displayed in the Output tab.

Residual by Predicted plot

You can use this plot either to identify outliers, or to diagnose non-linearity or non constant error variance. An ideal plot will show the points randomly scattered about the zero line.

The expected pattern is that the distribution of standardized deviance residuals across predicted values of the linear predictor has a mean value of zero and a constant range. The expected pattern is a horizontal line through zero.

GLE model nugget settings

On the Settings tab for a GLE model nugget, you specify options for raw propensity and for SQL generation during model scoring. This tab is available only after the model nugget is added to a stream.

Calculate raw propensity scores For models with flag targets only, you can request raw propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to standard prediction and confidence values. Adjusted propensity scores are not available.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL is generated:

- **Default:** Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL by using the scoring adapter and associated user-defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Cox Node

Cox regression builds a predictive model for time-to-event data. The model produces a survival function that predicts the probability that the event of interest has occurred at a given time t for given values of the predictor variables. The shape of the survival function and the regression coefficients for the predictors are estimated from observed subjects; the model can then be applied to new cases that have measurements for the predictor variables. Note that information from censored subjects, that is, those that do not experience the event of interest during the time of observation, contributes usefully to the estimation of the model.

Example. As part of its efforts to reduce customer churn, a telecommunications company is interested in modeling the "time to churn" in order to determine the factors that are associated with customers who are quick to switch to another service. To this end, a random sample of customers is selected, and their time spent as customers (whether or not they are still active customers) and various demographic fields are pulled from the database.

Requirements. You need one or more input fields, exactly one target field, and you must specify a survival time field within the Cox node. The target field should be coded so that the "false" value indicates survival and the "true" value indicates that the event of interest has occurred; it must have a measurement level of *Flag*, with string or integer storage. (Storage can be converted using a Filler or Derive node if necessary.) Fields set to *Both* or *None* are ignored. Fields used in the model must have their types fully instantiated. The survival time can be any numeric field.

Note: On scoring a Cox Regression model, an error is reported if empty strings in categorical variables are used as input to model building. Avoid using empty strings as input.

Dates & Times. Date & Time fields cannot be used to directly define the survival time; if you have Date & Time fields, you should use them to create a field containing survival times, based upon the difference between the date of entry into the study and the observation date.

Kaplan-Meier Analysis. Cox regression can be performed with no input fields. This is equivalent to a Kaplan-Meier analysis.

- [Cox Node Fields Options](#)
- [Cox Node Model Options](#)
- [Cox Node Expert Options](#)
- [Cox Node Settings Options](#)
- [Cox Model Nugget](#)

Related information

- [Overview of modeling nodes](#)
- [Modeling Node Fields Options](#)
- [Cox Node Model Options](#)
- [Cox Node Expert Options](#)
- [Cox Node Convergence Criteria](#)
- [Cox Node Advanced Output Options](#)
- [Cox Node Stepping Criteria](#)

- [Cox Node Settings Options](#)

Cox Node Fields Options

Survival time. Choose a numeric field (one with a measurement level of *Continuous*) in order to make the node executable. Survival time indicates the lifespan of the record being predicted. For example, when modeling customer time to churn, this would be the field that records how long the customer has been with the organization. The date on which the customer joined or churned would not affect the model; only the duration of the customer's tenure would be relevant.

Survival time is taken to be a duration with no units. You must make sure that the input fields match the survival time. For example, in a study to measure churn by months, you would use sales per month as an input instead of sales per year. If your data has start and end dates instead of a duration, you must recode those dates to a duration upstream from the Cox node.

The remaining fields in this dialog box are the standard ones used throughout IBM® SPSS® Modeler. See the topic [Modeling Node Fields Options](#) for more information.

Cox Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Create split models. Builds a separate model for each possible value of input fields that are specified as split fields. See [Building Split Models](#) for more information.

Method. The following options are available for entering predictors into the model:

- **Enter.** This is the default method, which enters all of the terms into the model directly. No field selection is performed in building the model.
- **Stepwise.** The Stepwise method of field selection builds the model in steps, as the name implies. The initial model is the simplest model possible, with no model terms (except the constant) in the model. At each step, terms that have not yet been added to the model are evaluated, and if the best of those terms adds significantly to the predictive power of the model, it is added. In addition, terms that are currently in the model are reevaluated to determine if any of them can be removed without significantly detracting from the model. If so, they are removed. The process repeats, and other terms are added and/or removed. When no more terms can be added to improve the model, and no more terms can be removed without detracting from the model, the final model is generated.
- **Backwards Stepwise.** The Backwards Stepwise method is essentially the opposite of the Stepwise method. With this method, the initial model contains all of the terms as predictors. At each step, terms in the model are evaluated, and any terms that can be removed without significantly detracting from the model are removed. In addition, previously removed terms are reevaluated to determine if the best of those terms adds significantly to the predictive power of the model. If so, it is added back into the model. When no more terms can be removed without significantly detracting from the model, and no more terms can be added to improve the model, the final model is generated.

Note: The automatic methods, including Stepwise and Backwards Stepwise, are highly adaptable learning methods and have a strong tendency to overfit the training data. When using these methods, it is especially important to verify the validity of the resulting model either with new data or a hold-out test sample created using the Partition node.

Groups. Specifying a groups field causes the node to compute separate models for each category of the field. It can be any categorical field (Flag or Nominal) with string or integer storage.

Model type. There are two options for defining the terms in the model. **Main effects** models include only the input fields individually and do not test interactions (multiplicative effects) between input fields. **Custom** models include only the terms (main effects and interactions) that you specify. When selecting this option, use the Model Terms list to add or remove terms in the model.

Model Terms. When building a Custom model, you will need to explicitly specify the terms in the model. The list shows the current set of terms for the model. The buttons on the right side of the Model Terms list allow you to add and remove model terms.

- To add terms to the model, click the *Add new model terms* button. See the topic [Adding Terms to a Cox Regression Model](#) for more information.
- To delete terms, select the desired terms and click the *Delete selected model terms* button.
- [Adding Terms to a Cox Regression Model](#)

Related information

- [Cox Node](#)
 - [Cox Node Expert Options](#)
 - [Cox Node Convergence Criteria](#)
 - [Cox Node Advanced Output Options](#)
 - [Cox Node Stepping Criteria](#)
 - [Cox Node Settings Options](#)
-

Adding Terms to a Cox Regression Model

When requesting a custom model, you can add terms to the model by clicking the *Add new model terms* button on the Model tab. A new dialog box opens in which you can specify terms.

Type of term to add. There are several ways to add terms to the model, based on the selection of input fields in the Available fields list.

- **Single interaction.** Inserts the term representing the interaction of all selected fields.
- **Main effects.** Inserts one main effect term (the field itself) for each selected input field.
- **All 2-way interactions.** Inserts a 2-way interaction term (the product of the input fields) for each possible pair of selected input fields. For example, if you have selected input fields *A*, *B*, and *C* in the Available fields list, this method will insert the terms *A * B*, *A * C*, and *B * C*.
- **All 3-way interactions.** Inserts a 3-way interaction term (the product of the input fields) for each possible combination of selected input fields, taken three at a time. For example, if you have selected input fields *A*, *B*, *C*, and *D* in the Available fields list, this method will insert the terms *A * B * C*, *A * B * D*, *A * C * D*, and *B * C * D*.
- **All 4-way interactions.** Inserts a 4-way interaction term (the product of the input fields) for each possible combination of selected input fields, taken four at a time. For example, if you have selected input fields *A*, *B*, *C*, *D*, and *E* in the Available fields list, this method will insert the terms *A * B * C * D*, *A * B * C * E*, *A * B * D * E*, *A * C * D * E*, and *B * C * D * E*.

Available fields. Lists the available input fields to be used in constructing model terms. Note that the list may include fields that are not legal input fields, so take care to ensure that all model terms include only input fields.

Preview. Shows the terms that will be added to the model if you click Insert, based on the selected fields and the term type selected above.

Insert. Inserts terms in the model (based on the current selection of fields and term type) and closes the dialog box.

Cox Node Expert Options

Convergence. These options allow you to control the parameters for model convergence. When you execute the model, the convergence settings control how many times the different parameters are repeatedly run through to see how well they fit. The more often the parameters are tried, the closer the results will be (that is, the results will converge). See the topic [Cox Node Convergence Criteria](#) for more information.

Output. These options allow you to request additional statistics and plots, including the survival curve, that will be displayed in the advanced output of the generated model built by the node. See the topic [Cox Node Advanced Output Options](#) for more information.

Stepping. These options allow you to control the criteria for adding and removing fields with the Stepwise estimation method. (The button is disabled if the Enter method is selected.) See the topic [Cox Node Stepping Criteria](#) for more information.

- [Cox Node Convergence Criteria](#)
- [Cox Node Advanced Output Options](#)
- [Cox Node Stepping Criteria](#)

Related information

- [Cox Node](#)
 - [Cox Node Model Options](#)
 - [Cox Node Convergence Criteria](#)
 - [Cox Node Advanced Output Options](#)
 - [Cox Node Stepping Criteria](#)
 - [Cox Node Settings Options](#)
-

Cox Node Convergence Criteria

Maximum iterations. Allows you to specify the maximum iterations for the model, which controls how long the procedure will search for a solution.

Log-likelihood convergence. Iterations stop if the relative change in the log-likelihood is less than this value. The criterion is not used if the value is 0.

Parameter convergence. Iterations stop if the absolute change or relative change in the parameter estimates is less than this value. The criterion is not used if the value is 0.

Related information

- [Cox Node](#)
 - [Cox Node Model Options](#)
 - [Cox Node Expert Options](#)
 - [Cox Node Advanced Output Options](#)
 - [Cox Node Stepping Criteria](#)
 - [Cox Node Settings Options](#)
-

Cox Node Advanced Output Options

Statistics. You can obtain statistics for your model parameters, including confidence intervals for $\exp(B)$ and correlation of estimates. You can request these statistics either at each step or at the last step only.

Display baseline function. Allows you to display the baseline hazard function and cumulative survival at the mean of the covariates.

Plots

Plots can help you to evaluate your estimated model and interpret the results. You can plot the survival, hazard, log-minus-log, and one-minus-survival functions.

- **Survival.** Displays the cumulative survival function on a linear scale.
- **Hazard.** Displays the cumulative hazard function on a linear scale.
- **Log minus log.** Displays the cumulative survival estimate after the $\ln(-\ln)$ transformation is applied to the estimate.
- **One minus survival.** Plots one-minus the survival function on a linear scale.

Plot a separate line for each value. This option is available only for categorical fields.

Value to use for plots. Because these functions depend on values of the predictors, you must use constant values for the predictors to plot the functions versus time. The default is to use the mean of each predictor as a constant value, but you can enter your own values for the plot using the grid. For categorical inputs, indicator coding is used, so there is a regression coefficient for each category (except the last). Thus, a categorical input has a mean value for each indicator contrast, equal to the proportion of cases in the category corresponding to the indicator contrast.

Related information

- [Cox Node](#)
 - [Cox Node Model Options](#)
 - [Cox Node Expert Options](#)
 - [Cox Node Convergence Criteria](#)
 - [Cox Node Stepping Criteria](#)
 - [Cox Node Settings Options](#)
-

Cox Node Stepping Criteria

Removal criterion. Select Likelihood Ratio for a more robust model. To shorten the time required to build the model, you can try selecting Wald. There is the additional option Conditional, which provides removal testing based on the probability of the likelihood-ratio statistic based on conditional parameter estimates.

Significance thresholds for criteria. This option allows you to specify selection criteria based on the statistical probability (the p value) associated with each field. Fields will be added to the model only if the associated p value is smaller than the Entry value and will be removed only if the p value is larger than the Removal value. The Entry value must be smaller than the Removal value.

Related information

- [Cox Node](#)
- [Cox Node Model Options](#)
- [Cox Node Expert Options](#)

- [Cox Node Convergence Criteria](#)
 - [Cox Node Advanced Output Options](#)
 - [Cox Node Settings Options](#)
-

Cox Node Settings Options

Predict survival at future times. Specify one or more future times. Survival, that is, whether each case is likely to survive for at least that length of time (from now) without the terminal event occurring, is predicted for each record at each time value, one prediction per time value. Note that survival is the "false" value of the target field.

- **Regular intervals.** Survival time values are generated from the specified Time interval and Number of time periods to score. For example, if 3 time periods are requested with an interval of 2 between each time, survival will be predicted for future times 2, 4, 6. Every record is evaluated at the same time values.
- **Time fields.** Survival times are provided for each record in the time field chosen (one prediction field is generated), thus each record can be evaluated at different times.

Past survival time. Specify the survival time of the record so far—for example, the tenure of an existing customer as a field. Scoring the likelihood of survival at a future time will be conditional on past survival time.

Note: The values of future and past survival times must be within range of survival times in the data used to train the model. Records whose times fall outside this range are scored as null.

Append all probabilities. Specifies whether probabilities for each category of the output field are added to each record processed by the node. If this option is not selected, the probability of only the predicted category is added. Probabilities are computed for each future time.

Calculate cumulative hazard function. Specifies whether the value of the cumulative hazard is added to each record. The cumulative hazard is computed for each future time.

Related information

- [Cox Node](#)
 - [Cox Node Model Options](#)
 - [Cox Node Expert Options](#)
 - [Cox Node Convergence Criteria](#)
 - [Cox Node Advanced Output Options](#)
 - [Cox Node Stepping Criteria](#)
-

Cox Model Nugget

Cox regression models represent the equations estimated by Cox nodes. They contain all of the information captured by the model, as well as information about the model structure and performance.

When you run a stream containing a generated Cox regression model, the node adds two new fields containing the model's prediction and the associated probability. The names of the new fields are derived from the name of the output field being predicted, prefixed with \$C- for the predicted category and \$CP- for the associated probability, suffixed with the number of the future time interval or the name of the time field that defines the time interval. For example, for an output field named *churn* and two future time intervals defined at regular intervals, the new fields would be named \$C-churn-1, \$CP-churn-1, \$C-churn-2, and \$CP-churn-2. If future times are defined with a time field *tenure*, the new fields would be \$C-churn_tenure and \$CP-churn_tenure.

If you have selected the Append all probabilities settings option in the Cox node, two additional fields will be added for each future time, containing the probabilities of survival and failure for each record. These additional fields are named based on the name of the output field, prefixed by \$CP-<false value>- for the probability of survival and \$CP-<true value>- for the probability the event has occurred, suffixed with the number of the future time interval. For example, for an output field where the "false" value is 0 and the "true" value is 1, and two future time intervals defined at regular intervals, the new fields would be named \$CP-0-1, \$CP-1-1, \$CP-0-2, and \$CP-1-2. If future times are defined with a single time field *tenure*, the new fields would be \$CP-0-1 and \$CP-1-1, since there is a single future interval

If you have selected the Calculate cumulative hazard function settings option in the Cox Node, an additional field will be added for each future time, containing the cumulative hazard function for each record. These additional fields are named based on the name of the output field, prefixed by \$CH-, suffixed with the number of the future time interval or the name of the time field that defines the time interval. For example, for an output field named *churn* and two future time intervals defined at regular intervals, the new fields would be named \$CH-churn-1 and \$CH-churn-2. If future times are defined with a time field *tenure*, the new field would be \$CH-churn-1.

- [Cox Regression Output Settings](#)
- [Cox Regression Advanced Output](#)

Related information

- [Cox Regression Output Settings](#)
 - [Cox Regression Advanced Output](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
 - [Logistic Node](#)
-

Cox Regression Output Settings

Except for SQL generation, the Settings tab of the nugget contains the same controls as the Settings tab of the model node. The default values of the nugget controls are determined by the values set in the model node. See the topic [Cox Node Settings Options](#) for more information.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Related information

- [Cox Model Nugget](#)
 - [Cox Regression Advanced Output](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Cox Regression Advanced Output

The advanced output for Cox regression gives detailed information about the estimated model and its performance, including the survival curve. Most of the information contained in the advanced output is quite technical, and extensive knowledge of Cox regression is required to properly interpret this output.

Related information

- [Cox Model Nugget](#)
 - [Cox Regression Output Settings](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Clustering models

Clustering models focus on identifying groups of similar records and labeling the records according to the group to which they belong. This is done without the benefit of prior knowledge about the groups and their characteristics. In fact, you may not even know exactly how many groups to look for. This is what distinguishes clustering models from the other machine-learning techniques—there is no predefined output or target field for the model to predict. These models are often referred to as **unsupervised learning** models, since there is no external standard by which to judge the model's classification performance. There are no *right* or *wrong* answers for these models. Their value is determined by their ability to capture interesting groupings in the data and provide useful descriptions of those groupings.

Clustering methods are based on measuring distances between records and between clusters. Records are assigned to clusters in a way that tends to minimize the distance between records belonging to the same cluster.

The following clustering methods are provided:

	The K-Means node clusters the data set into distinct groups (or clusters). The method defines a fixed number of clusters, iteratively assigns records to clusters, and adjusts the cluster centers until further refinement can no longer improve the model. Instead of trying to predict an outcome, <i>k</i> -means uses a process known as unsupervised learning to uncover patterns in the set of input fields.
	The TwoStep node uses a two-step clustering method. The first step makes a single pass through the data to compress the raw input data into a manageable set of subclusters. The second step uses a hierarchical clustering method to progressively merge the subclusters into larger and larger clusters. TwoStep has the advantage of automatically estimating the optimal number of clusters for the training data. It can handle mixed field types and large data sets efficiently.
	The Kohonen node generates a type of neural network that can be used to cluster the data set into distinct groups. When the network is fully trained, records that are similar should be close together on the output map, while records that are different will be far apart. You can look at the number of observations captured by each unit in the model nugget to identify the strong units. This may give you a sense of the appropriate number of clusters.
	Hierarchical Density-Based Spatial Clustering (HDBSCAN)® uses unsupervised learning to find clusters, or dense regions, of a data set. The HDBSCAN node in SPSS® Modeler exposes the core features and commonly used parameters of the HDBSCAN library. The node is implemented in Python, and you can use it to cluster your dataset into distinct groups when you don't know what those groups are at first.

Clustering models are often used to create clusters or segments that are then used as inputs in subsequent analyses. A common example of this is the market segments used by marketers to partition their overall market into homogeneous subgroups. Each segment has special characteristics that affect the success of marketing efforts targeted toward it. If you are using data mining to optimize your marketing strategy, you can usually improve your model significantly by identifying the appropriate segments and using that segment information in your predictive models.

- [**Kohonen node**](#)
- [**Kohonen Model Nuggets**](#)
- [**K-Means Node**](#)
- [**K-Means Model Nuggets**](#)
- [**TwoStep Cluster node**](#)
- [**TwoStep Cluster Model Nuggets**](#)
- [**TwoStep-AS Cluster node**](#)
- [**TwoStep-AS Cluster Model Nuggets**](#)
- [**K-Means-AS node**](#)
- [**The Cluster Viewer**](#)

Kohonen node

Kohonen networks are a type of neural network that perform clustering, also known as a **knet** or a **self-organizing map**. This type of network can be used to cluster the dataset into distinct groups when you don't know what those groups are at the beginning. Records are grouped so that records within a group or cluster tend to be similar to each other, and records in different groups are dissimilar.

The basic units are **neurons**, and they are organized into two layers: the **input layer** and the **output layer** (also called the **output map**). All of the input neurons are connected to all of the output neurons, and these connections have **strengths**, or **weights**, associated with them. During training, each unit competes with all of the others to "win" each record.

The output map is a two-dimensional grid of neurons, with no connections between the units.

Input data is presented to the input layer, and the values are propagated to the output layer. The output neuron with the strongest response is said to be the **winner** and is the answer for that input.

Initially, all weights are random. When a unit wins a record, its weights (along with those of other nearby units, collectively referred to as a **neighborhood**) are adjusted to better match the pattern of predictor values for that record. All of the input records are shown, and weights are updated accordingly. This process is repeated many times until the changes become very small. As training proceeds, the weights on the grid units are adjusted so that they form a two-dimensional "map" of the clusters (hence the term **self-organizing map**).

When the network is fully trained, records that are similar should be close together on the output map, whereas records that are vastly different will be far apart.

Unlike most learning methods in IBM® SPSS® Modeler, Kohonen networks do *not* use a target field. This type of learning, with no target field, is called **unsupervised learning**. Instead of trying to predict an outcome, Kohonen nets try to uncover patterns in the set of input fields. Usually, a Kohonen net will end up with a few units that summarize many observations (**strong** units), and several units that don't really correspond to any of the observations (**weak** units). The strong units (and sometimes other units adjacent to them in the grid) represent probable cluster centers.

Another use of Kohonen networks is in **dimension reduction**. The spatial characteristic of the two-dimensional grid provides a mapping from the k original predictors to two derived features that preserve the similarity relationships in the original predictors. In some cases, this can give you the same kind of benefit as factor analysis or PCA.

Requirements. To train a Kohonen net, you need one or more fields with the role set to *Input*. Fields with the role set to *Target*, *Both*, or *None* are ignored.

Strengths. You do not need to have data on group membership to build a Kohonen network model. You don't even need to know the number of groups to look for. Kohonen networks start with a large number of units, and as training progresses, the units gravitate toward the natural clusters in the data. You can look at the number of observations captured by each unit in the model nugget to identify the strong units, which can give you a sense of the appropriate number of clusters.

- [Kohonen Node Model Options](#)
 - [Kohonen Node Expert Options](#)
-

Kohonen Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Continue training existing model. By default, each time you execute a Kohonen node, a completely new network is created. If you select this option, training continues with the last net successfully produced by the node.

Show feedback graph. If this option is selected, a visual representation of the two-dimensional array is displayed during training. The strength of each node is represented by color. Red denotes a unit that is winning many records (a **strong** unit), and white denotes a unit that is winning few or no records (a **weak** unit). Feedback may not display if the time taken to build the model is relatively short. Note that this feature can slow training time. To speed up training time, deselect this option.

Stop on. The default stopping criterion stops training, based on internal parameters. You can also specify time as the stopping criterion. Enter the time (in minutes) for the network to train.

Set random seed. If no random seed is set, the sequence of random values used to initialize the network weights will be different every time the node is executed. This can cause the node to create different models on different runs, even if the node settings and data values are exactly the same. By selecting this option, you can set the random seed to a specific value so the resulting model is exactly reproducible. A specific random seed always generates the same sequence of random values, in which case executing the node always yields the same generated model.

Note: When using the Set random seed option with records read from a database, a Sort node may be required prior to sampling in order to ensure the same result each time the node is executed. This is because the random seed depends on the order of records, which is not guaranteed to stay the same in a relational database.

Note: If you want to include nominal (set) fields in your model but are having memory problems in building the model, or the model is taking too long to build, consider recoding large set fields to reduce the number of values, or consider using a different field with fewer values as a proxy for the large set. For example, if you are having a problem with a *product_id* field containing values for individual products, you might consider removing it from the model and adding a less detailed *product_category* field instead.

Optimize. Select options designed to increase performance during model building based on your specific needs.

- Select Speed to instruct the algorithm to never use disk spilling in order to improve performance.
- Select Memory to instruct the algorithm to use disk spilling when appropriate at some sacrifice to speed. This option is selected by default.
Note: When running in distributed mode, this setting can be overridden by administrator options specified in options.cfg.

Append cluster label. Selected by default for new models, but deselected for models loaded from earlier versions of IBM® SPSS® Modeler, this creates a single categorical score field of the same type that is created by both the K-Means and TwoStep nodes. This string field is used in the Auto Cluster node when calculating ranking measures for the different model types. See the topic [Auto Cluster node](#) for more information.

Related information

- [Kohonen node](#)
 - [Kohonen Node Expert Options](#)
-

Kohonen Node Expert Options

For those with detailed knowledge of Kohonen networks, expert options allow you to fine-tune the training process. To access expert options, set the Mode to Expert on the Expert tab.

Width and Length. Specify the size (width and length) of the two-dimensional output map as number of output units along each dimension.

Learning rate decay. Select either linear or exponential learning rate decay. The **learning rate** is a weighting factor that decreases over time, such that the network starts off encoding large-scale features of the data and gradually focuses on more fine-level detail.

Phase 1 and Phase 2. Kohonen net training is split into two phases. Phase 1 is a rough estimation phase, used to capture the gross patterns in the data. Phase 2 is a tuning phase, used to adjust the map to model the finer features of the data. For each phase, there are three parameters:

- **Neighborhood.** Sets the starting size (radius) of the neighborhood. This determines the number of "nearby" units that get updated along with the winning unit during training. During phase 1, the neighborhood size starts at *Phase 1 Neighborhood* and decreases to (*Phase 2 Neighborhood* + 1). During phase 2, neighborhood size starts at *Phase 2 Neighborhood* and decreases to 1.0. *Phase 1 Neighborhood* should be larger than *Phase 2 Neighborhood*.
- **Initial Eta.** Sets the starting value for learning rate **eta**. During phase 1, eta starts at *Phase 1 Initial Eta* and decreases to *Phase 2 Initial Eta*. During phase 2, eta starts at *Phase 2 Initial Eta* and decreases to 0. *Phase 1 Initial Eta* should be larger than *Phase 2 Initial Eta*.
- **Cycles.** Sets the number of cycles for each phase of training. Each phase continues for the specified number of passes through the data.

Related information

- [Kohonen node](#)
 - [Kohonen Node Model Options](#)
-

Kohonen Model Nuggets

Kohonen model nuggets contain all of the information captured by the trained Kohonen network, as well as information about the network's architecture.

When you run a stream containing a Kohonen model nugget, the node adds two new fields containing the X and Y coordinates of the unit in the Kohonen output grid that responded most strongly to that record. The new field names are derived from the model name, prefixed by \$KX- and \$KY-. For example, if your model is named *Kohonen*, the new fields would be named \$KX-*Kohonen* and \$KY-*Kohonen*.

To get a better sense of what the Kohonen net has encoded, click the Model tab on the model nugget browser. This displays the Cluster Viewer, providing a graphical representation of clusters, fields, and importance levels. See the topic [Cluster Viewer - Model Tab](#) for more information.

If you prefer to visualize the clusters as a grid, you can view the result of the Kohonen net by plotting the \$KX- and \$KY- fields using a Plot node. (You should select X-Agitation and Y-Agitation in the Plot node to prevent each unit's records from all being plotted on top of each other.) In the plot, you can also overlay a symbolic field to investigate how the Kohonen net has clustered the data.

Another powerful technique for gaining insight into the Kohonen network is to use rule induction to discover the characteristics that distinguish the clusters found by the network.

For general information on using the model browser, see [Browsing model nuggets](#)

- [Kohonen Model Summary](#)

Related information

- [Kohonen Model Summary](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
 - [Kohonen node](#)
 - [K-Means Model Nuggets](#)
 - [TwoStep Cluster Model Nuggets](#)
 - [The Cluster Viewer](#)
 - [Cluster Viewer - Model Tab](#)
-

Kohonen Model Summary

The Summary tab for a Kohonen model nugget displays information about the architecture or topology of the network. The length and width of the two-dimensional Kohonen feature map (the output layer) are shown as \$KX- *model_name* and \$KY- *model_name*. For the input and output layers, the number of units in that layer is listed.

Related information

- [Kohonen Model Nuggets](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
 - [Cluster Viewer - Model Tab](#)
-

K-Means Node

The K-Means node provides a method of **cluster analysis**. It can be used to cluster the dataset into distinct groups when you don't know what those groups are at the beginning. Unlike most learning methods in IBM® SPSS® Modeler, K-Means models do *not* use a target field. This type of learning, with no target field, is called **unsupervised learning**. Instead of trying to predict an outcome, K-Means tries to uncover patterns in the set of input fields. Records are grouped so that records within a group or cluster tend to be similar to each other, but records in different groups are dissimilar.

K-Means works by defining a set of starting cluster centers derived from data. It then assigns each record to the cluster to which it is most similar, based on the record's input field values. After all cases have been assigned, the cluster centers are updated to reflect the new set of records assigned to each cluster. The records are then checked again to see whether they should be reassigned to a different cluster, and the record assignment/cluster iteration process continues until either the maximum number of iterations is reached, or the change between one iteration and the next fails to exceed a specified threshold.

Note: The resulting model depends to a certain extent on the order of the training data. Reordering the data and rebuilding the model may lead to a different final cluster model.

Requirements. To train a K-Means model, you need one or more fields with the role set to *Input*. Fields with the role set to *Output*, *Both*, or *None* are ignored.

Strengths. You do not need to have data on group membership to build a K-Means model. The K-Means model is often the fastest method of clustering for large datasets.

- [K-Means Node Model Options](#)
- [K-Means Node Expert Options](#)

Related information

- [Overview of modeling nodes](#)
 - [Modeling Node Fields Options](#)
 - [K-Means Node Model Options](#)
 - [K-Means Node Expert Options](#)
 - [TwoStep Cluster node](#)
 - [Kohonen node](#)
-

K-Means Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Specified number of clusters. Specify the number of clusters to generate. The default is 5.

Generate distance field. If this option is selected, the model nugget will include a field containing the distance of each record from the center of its assigned cluster.

Cluster label. Specify the format for the values in the generated cluster membership field. Cluster membership can be indicated as a String with the specified Label prefix (for example "Cluster 1", "Cluster 2", and so on), or as a Number.

Note: If you want to include nominal (set) fields in your model but are having memory problems in building the model or the model is taking too long to build, consider recoding large set fields to reduce the number of values, or consider using a different field with fewer values as a proxy for the large set. For example, if you are having a problem with a *product_id* field containing values for individual products, you might consider removing it from the model and adding a less detailed *product_category* field instead.

Optimize. Select options designed to increase performance during model building based on your specific needs.

- Select Speed to instruct the algorithm to never use disk spilling in order to improve performance.
- Select Memory to instruct the algorithm to use disk spilling when appropriate at some sacrifice to speed. This option is selected by default.
Note: When running in distributed mode, this setting can be overridden by administrator options specified in options.cfg.

Related information

- [K-Means Node](#)
- [K-Means Node Expert Options](#)

K-Means Node Expert Options

For those with detailed knowledge of *k*-means clustering, expert options allow you to fine-tune the training process. To access expert options, set the Mode to Expert on the Expert tab.

Stop on. Specify the stopping criterion to be used in training the model. The Default stopping criterion is 20 iterations or change < 0.000001, whichever occurs first. Select Custom to specify your own stopping criteria.

- **Maximum Iterations.** This option allows you to stop model training after the number of iterations specified.
- **Change tolerance.** This option allows you to stop model training when the largest change in cluster centers for an iteration is less than the level specified.

Encoding value for sets. Specify a value between 0 and 1.0 to use for recoding set fields as groups of numeric fields. The default value is the square root of 0.5 (approximately 0.707107), which provides the proper weighting for recoded flag fields. Values closer to 1.0 will weight set fields more heavily than numeric fields.

Related information

- [K-Means Node](#)
- [K-Means Node Model Options](#)

K-Means Model Nuggets

K-Means model nuggets contain all of the information captured by the clustering model, as well as information about the training data and the estimation process.

When you run a stream containing a K-Means modeling node, the node adds two new fields containing the cluster membership and distance from the assigned cluster center for that record. The new field names are derived from the model name, prefixed by \$KM- for the cluster membership and \$KMD- for the distance from the cluster center. For example, if your model is named *Kmeans*, the new fields would be named \$KM-Kmeans and \$KMD-Kmeans.

A powerful technique for gaining insight into the K-Means model is to use rule induction to discover the characteristics that distinguish the clusters found by the model. You can also click the Model tab on the model nugget browser to display the Cluster Viewer, providing a graphical representation of clusters, fields, and importance levels. See the topic [Cluster Viewer - Model Tab](#) for more information.

For general information on using the model browser, see [Browsing model nuggets](#)

- [K-Means Model Summary](#)

Related information

- [K-Means Model Summary](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)
- [K-Means Node](#)
- [Kohonen Model Nuggets](#)
- [TwoStep Cluster Model Nuggets](#)
- [The Cluster Viewer](#)
- [Cluster Viewer - Model Tab](#)

K-Means Model Summary

The Summary tab for a K-Means model nugget contains information about the training data, the estimation process, and the clusters defined by the model. The number of clusters is shown, as well as the iteration history. If you have executed an Analysis node attached to this modeling node, information from that analysis will also be displayed in this section.

Related information

- [K-Means Model Nuggets](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)
- [Cluster Viewer - Model Tab](#)

TwoStep Cluster node

The TwoStep Cluster node provides a form of **cluster analysis**. It can be used to cluster the dataset into distinct groups when you don't know what those groups are at the beginning. As with Kohonen nodes and K-Means nodes, TwoStep Cluster models do *not* use a target field. Instead of trying to predict an outcome, TwoStep Cluster tries to uncover patterns in the set of input fields. Records are grouped so that records within a group or cluster tend to be similar to each other, but records in different groups are dissimilar.

TwoStep Cluster is a two-step clustering method. The first step makes a single pass through the data, during which it compresses the raw input data into a manageable set of subclusters. The second step uses a hierarchical clustering method to progressively merge the subclusters into larger and larger clusters, without requiring another pass through the data. Hierarchical clustering has the advantage of not requiring the number of clusters to be selected ahead of time. Many hierarchical clustering methods start with individual records as starting clusters and merge them recursively to produce ever larger clusters. Though such approaches often break down with large amounts of data, TwoStep's initial preclustering makes hierarchical clustering fast even for large datasets.

Note: The resulting model depends to a certain extent on the order of the training data. Reordering the data and rebuilding the model may lead to a different final cluster model.

Requirements. To train a TwoStep Cluster model, you need one or more fields with the role set to *Input*. Fields with the role set to *Target*, *Both*, or *None* are ignored. The TwoStep Cluster algorithm does not handle missing values. Records with blanks for any of the input fields will be ignored when building the model.

Strengths. TwoStep Cluster can handle mixed field types and is able to handle large datasets efficiently. It also has the ability to test several cluster solutions and choose the best, so you don't need to know how many clusters to ask for at the outset. TwoStep Cluster can be set to automatically exclude **outliers**, or extremely unusual cases that can contaminate your results.

IBM® SPSS® Modeler has two different versions of the TwoStep Cluster node:

- TwoStep Cluster is the traditional node that runs on the IBM SPSS Modeler Server.
- TwoStep-AS Cluster can run when connected to IBM SPSS Analytic Server.
- [TwoStep Cluster Node Model Options](#)

TwoStep Cluster Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Standardize numeric fields. By default, TwoStep will standardize all numeric input fields to the same scale, with a mean of 0 and a variance of 1. To retain the original scaling for numeric fields, deselect this option. Symbolic fields are not affected.

Exclude outliers. If you select this option, records that don't seem to fit into a substantive cluster will be automatically excluded from the analysis. This prevents such cases from distorting the results.

Outlier detection occurs during the preclustering step. When this option is selected, subclusters with few records relative to other subclusters are considered potential outliers, and the tree of subclusters is rebuilt excluding those records. The size below which subclusters are considered to contain potential outliers is controlled by the Percentage option. Some of those potential outlier records can be added to the rebuilt

subclusters if they are similar enough to any of the new subcluster profiles. The rest of the potential outliers that cannot be merged are considered outliers and are added to a "noise" cluster and excluded from the hierarchical clustering step.

When scoring data with a TwoStep model that uses outlier handling, new cases that are more than a certain threshold distance (based on the log-likelihood) from the nearest substantive cluster are considered outliers and are assigned to the "noise" cluster with the name -1.

Cluster label. Specify the format for the generated cluster membership field. Cluster membership can be indicated as a String with the specified Label prefix (for example, "Cluster 1", "Cluster 2", and so on) or as a Number.

Automatically calculate number of clusters. TwoStep cluster can very rapidly analyze a large number of cluster solutions to choose the optimal number of clusters for the training data. Specify a range of solutions to try by setting the Maximum and the Minimum number of clusters.

Specify number of clusters. If you know how many clusters to include in your model, select this option and enter the number of clusters.

Distance measure. This selection determines how the similarity between two clusters is computed.

- **Log-likelihood.** The likelihood measure places a probability distribution on the variables. Continuous variables are assumed to be normally distributed, while categorical variables are assumed to be multinomial. All variables are assumed to be independent.
- **Euclidean.** The Euclidean measure is the "straight line" distance between two clusters. It can be used only when all of the variables are continuous.

Clustering Criterion. This selection determines how the automatic clustering algorithm determines the number of clusters. Either the Bayesian Information Criterion (BIC) or the Akaike Information Criterion (AIC) can be specified.

Related information

- [TwoStep Cluster node](#)
-

TwoStep Cluster Model Nuggets

TwoStep cluster model nuggets contain all of the information captured by the clustering model, as well as information about the training data and the estimation process.

When you run a stream containing a TwoStep cluster model nugget, the node adds a new field containing the cluster membership for that record. The new field name is derived from the model name, prefixed by \$T-. For example, if your model is named *TwoStep*, the new field would be named *\$T-TwoStep*.

A powerful technique for gaining insight into the TwoStep model is to use rule induction to discover the characteristics that distinguish the clusters found by the model. You can also click the Model tab on the model nugget browser to display the Cluster Viewer, providing a graphical representation of clusters, fields, and importance levels. See the topic [Cluster Viewer - Model Tab](#) for more information.

For general information on using the model browser, see [Browsing model nuggets](#)

- [TwoStep Model Summary](#)

Related information

- [TwoStep Model Summary](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
 - [TwoStep Cluster node](#)
 - [Kohonen Model Nuggets](#)
 - [K-Means Model Nuggets](#)
 - [The Cluster Viewer](#)
 - [Cluster Viewer - Model Tab](#)
-

TwoStep Model Summary

The Summary tab for a TwoStep cluster model nugget displays the number of clusters found, along with information about the training data, the estimation process, and build settings used.

See the topic [Browsing model nuggets](#) for more information.

Related information

- [TwoStep Cluster Model Nuggets](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
 - [Cluster Viewer - Model Tab](#)
-

TwoStep-AS Cluster node

IBM® SPSS® Modeler has two different versions of the TwoStep Cluster node:

- TwoStep Cluster is the traditional node that runs on the IBM SPSS Modeler Server.
 - TwoStep-AS Cluster can run when connected to IBM SPSS Analytic Server.
 - [Twostep-AS cluster analysis](#)
-

Twostep-AS cluster analysis

TwoStep Cluster is an exploratory tool that is designed to reveal natural groupings (or clusters) within a data set that would otherwise not be apparent. The algorithm that is employed by this procedure has several desirable features that differentiate it from traditional clustering techniques:

- Handling of categorical and continuous variables. By assuming variables to be independent, a joint multinomial-normal distribution can be placed on categorical and continuous variables.
- Automatic selection of number of clusters. By comparing the values of a model-choice criterion across different clustering solutions, the procedure can automatically determine the optimal number of clusters.
- Scalability. By constructing a cluster feature (CF) tree that summarizes the records, the TwoStep algorithm can analyze large data files.

For example, retail and consumer product companies regularly apply clustering techniques to information that describes their customers' buying habits, gender, age, income level, and other attributes. These companies tailor their marketing and product development strategies to each consumer group to increase sales and build brand loyalty.

- [Fields tab \(Twostep-AS Cluster\)](#)
 - [Basics \(Twostep-AS Cluster\)](#)
 - [Feature Tree Criteria \(Twostep-AS Cluster\)](#)
 - [Standardize](#)
 - [Feature Selection](#)
 - [Model Output](#)
 - [Model Options](#)
-

Fields tab (Twostep-AS Cluster)

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. All fields with a defined role of Input are selected.

Use custom field assignments. Add and remove fields regardless of their defined role assignments. You can select fields with any role and move them in or out of the Predictors (Inputs) list.

Basics (Twostep-AS Cluster)

Number of Clusters

Determine automatically

The procedure determines the best number of clusters, within the specified range. The Minimum must be greater than 1. This is the default option.

Specify fixed

The procedure generates the specified number of clusters. The Number must be greater than 1.

Clustering Criterion

This selection controls how the automatic clustering algorithm determines the number of clusters.

Bayesian Information Criterion (BIC)

A measure for selecting and comparing models based on the -2 log likelihood. Smaller values indicate better models. The BIC also "penalizes" overparameterized models (complex models with a large number of inputs, for example), but more strictly than the AIC.

Akaike Information Criterion (AIC)

A measure for selecting and comparing models based on the -2 log likelihood. Smaller values indicate better models. The AIC "penalizes" overparameterized models (complex models with a large number of inputs, for example).

Automatic Clustering Method

If you select Determine automatically, choose from the following clustering methods used to automatically determine the number of clusters:

Use Clustering Criterion setting

Information criteria convergence is the ratio of information criteria corresponding to two current cluster solutions and the first cluster solution. The criterion used is the one selected in the Clustering Criterion group.

Distance jump

Distance jump is the ratio of distances corresponding to two consecutive cluster solutions.

Maximum

Combine results from the information criteria convergence method and the distance jump method to produce the number of clusters corresponding to the second jump.

Minimum

Combine results from the information criteria convergence method and the distance jump method to produce the number of clusters corresponding to the first jump.

Feature Importance Method

Feature Importance Method determines how important the features (fields) are in the cluster solution. The output includes information about overall feature importance and the importance of each feature field in each cluster. Features that do not meet a minimum threshold are excluded.

Use Clustering Criterion setting

This is the default method, based on the criterion that is selected in the Clustering Criterion group.

Effect size

Feature importance is based on effect size instead of significance values.

Feature Tree Criteria (Twostep-AS Cluster)

These settings determine how the cluster feature tree is built. By building a cluster feature tree and summarizing the records, the TwoStep algorithm can analyze large data files. In other words, TwoStep Cluster uses a cluster feature tree to build clusters, enabling it to process many cases.

Distance Measure

This selection determines how the similarity between two clusters is computed.

Log-likelihood

The likelihood measure places a probability distribution on the fields. Continuous fields are assumed to be normally distributed, while categorical fields are assumed to be multinomial. All fields are assumed to be independent.

Euclidean

The Euclidean measure is the "straight line" distance between two clusters. Squared Euclidean measure and the Ward method are used to compute similarity between clusters. It can be used only when all of the fields are continuous.

Outlier Clusters

Include outlier clusters

Include clusters for cases that are outliers from the regular clusters. If this option is not selected, all cases are included in regular clusters.

Number of cases in feature tree leaf is less than.

If the number of cases in the feature tree leaf is less than the specified value, the leaf is considered an outlier. The value must be an integer greater than 1. If you change this value, higher values are likely to result in more outlier clusters.

Top percentage of outliers.

When the cluster model is built, outliers are ranked by outlier strength. The outlier strength that is required to be in the top percentage of outliers is used as the threshold for determining whether a case is classified as an outlier. Higher values mean that more cases are classified as outliers. The value must be between 1 - 100.

Additional settings

Initial distance change threshold

The initial threshold that is used to grow the cluster feature tree. If insertion of a leaf into a leaf of the tree yields tightness less than this threshold, the leaf is not split. If the tightness exceeds this threshold, the leaf is split.

Leaf node maximum branches

The maximum number of child nodes that a leaf node can have.

Non-leaf node maximum branches

The maximum number of child nodes that a non-leaf node can have.

Maximum tree depth

The maximum number of levels that the cluster tree can have.

Adjustment weight on measurement level

Reduces the influence of categorical fields by increasing the weight for continuous fields. This value represents a denominator for reducing the weight for categorical fields. So a default of 6, for example, gives categorical fields a weight of 1/6.

Memory allocation

The maximum amount of memory in megabytes (MB) that the cluster algorithm uses. If the procedure exceeds this maximum, it uses the disk to store information that does not fit in memory.

Delayed split

Delay rebuilding of the cluster feature tree. The clustering algorithm rebuilds the cluster feature tree multiple times as it evaluates new cases. This option can improve performance by delaying that operation and reducing the number of times the tree is rebuilt.

Standardize

The clustering algorithm works with standardized continuous fields. By default, all continuous fields are standardized. To save some time and computational effort, you can move continuous fields that are already standardized to the Do not standardize list.

Feature Selection

On the Feature Selection screen, you can set rules that determine when fields are excluded. For example, you can exclude fields that have numerous missing values.

Rules for Excluding Fields

Percentage of missing values is greater than.

Fields with a percentage of missing values greater than the specified value are excluded from the analysis. The value must be a positive number greater than zero and less than 100.

Number of categories for categorical fields is greater than.

Categorical fields with more than the specified number of categories are excluded from the analysis. The value must be a positive integer greater than 1.

Fields with a Tendency Toward a Single Value

Coefficient of variation for continuous fields is less than.

Continuous fields with a coefficient of variation less than the specified value are excluded from the analysis. The coefficient of variation is the ratio of the standard deviation to the mean. Lower values tend to indicate less variation in values. The value must be between 0 and 1.

Percentage of cases in a single category for categorical fields is greater than.

Categorical fields with a percentage of cases in a single category greater than the specified value are excluded from the analysis. The value must be greater than 0 and less than 100.

Adaptive feature selection

This option runs an extra data pass to find and remove the least important fields.

Model Output

Model Building Summary

Model specifications

Summary of model specifications, number of clusters in the final model, and inputs (fields) included in the final model.

Record summary

Number and percentage of records (cases) included and excluded from the model.

Excluded inputs

For any fields not included in the final model, the reason the field was excluded.

Evaluation

Model quality

Table of goodness and importance for each cluster and overall model goodness of fit.

Feature importance bar chart

Bar chart of feature (field) importance across all clusters. Features (fields) with longer bars in the chart are more important than fields with shorter bars. They are also sorted in descending order of importance (the bar at the top is the most important).

Feature importance word cloud

Word cloud of feature (field) importance across all clusters. Features (fields) with larger text are more important than those with smaller text.

Outlier clusters

These options are disabled if you chose not to include outliers.

Interactive table and chart

Table and chart of outlier strength and the relative similarity of outlier clusters to regular clusters. Selecting different rows in the table displays information for different outlier clusters in the chart.

Pivot table

Table of outlier strength and the relative similarity of outlier clusters to regular clusters. This table contains the same information as the interactive display. This table supports all standard features for pivoting and editing tables.

Maximum number

The maximum number of outliers to display in the output. If there are more than twenty outlier clusters, a pivot table will be displayed instead.

Interpretation

Across cluster feature importance profiles

Interactive table and chart.

Table and charts of feature importance and cluster centers for each input (field) used in the cluster solution. Selecting different rows in the table displays a different chart. For categorical fields, a bar chart is displayed. For continuous fields, a chart of means and standard deviations is displayed.

Pivot table.

Table of feature importance and cluster centers for each input (field). This table contains the same information as the interactive display. This table supports all standard features for pivoting and editing tables.

Within cluster feature importance

For each cluster, the cluster center and feature importance for each input (field). There is a separate table for each cluster.

Cluster distances

A panel chart that displays the distances between clusters. There is a separate panel for each cluster.

Cluster label

Text

The label for each cluster is the value that is specified for Prefix, followed by a sequential number.

Number

The label for each cluster is a sequential number.

Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

TwoStep-AS Cluster Model Nuggets

The TwoStep-AS model nugget displays details of the model in the Model tab of the Output Viewer. For more information on using the viewer, see [Working with output](#).

The TwoStep-AS cluster model nuggets contain all of the information captured by the clustering model, as well as information about the training data and the estimation process.

When you run a stream containing a TwoStep-AS cluster model nugget, the node adds a new field containing the cluster membership for that record. The new field name is derived from the model name, prefixed by **\$AS-**. For example, if your model is named TwoStep, the new field will be named **\$AS-TwoStep**.

A powerful technique for gaining insight into the TwoStep-AS model is to use rule induction to discover the characteristics that distinguish the clusters found by the model.

For general information on using the model browser, see [Browsing model nuggets](#).

Note: In the TwoStep-AS model viewer, under the Evaluation > Model Quality section, the number of records is displayed for each cluster. If you connect a Distribution node to calculate the score result, you may find that the number of records in each cluster is different from what you see under Model Quality section. This functions as it should. From the algorithm perspective, the evaluation result comes from a hierarchical clustering process, while the scoring table comes from directly comparing the data case with the distribution of the final clusters. These two different scoring processes may result in different results if the clustering model is not perfect. But in most cases, the difference is quite small.

- [TwoStep-AS Cluster Model Nugget Settings](#)

TwoStep-AS Cluster Model Nugget Settings

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score by converting to native SQL If selected, generates native SQL to score the model within the database.
Note: Although this option can provide quicker results, the size and complexity of the native SQL increases as the complexity of the model increases.
- Score outside of the database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

K-Means-AS node

K-Means is one of the most commonly used clustering algorithms. It clusters data points into a predefined number of clusters.¹ The K-Means-AS node in SPSS® Modeler is implemented in Spark.

For details about K-Means algorithms, see <https://spark.apache.org/docs/2.2.0/ml-clustering.html>.

Note that the K-Means-AS node performs one-hot encoding automatically for categorical variables.

¹ "Clustering." Apache Spark. MLlib: Main Guide. Web. 3 Oct 2017.

- [K-Means-AS node Fields](#)
- [K-Means-AS node Build Options](#)
- [K-Means-AS node Fields](#)
- [K-Means-AS node Build Options](#)

K-Means-AS node Fields

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. This option tells the node to use field information from an upstream Type node. It's selected by default.

Use custom field assignments. If you want to manually assign input fields, select this option and then select the input field or fields. Using this option is similar to setting the field role in Input in a Type node.

K-Means-AS node Build Options

Use the Build Options tab to specify build options for the K-Means-AS node, including regular options for model building, initialization options for initializing cluster centers, and advanced options for the computing iteration and random seed. For more information, see the [JavaDoc for K-Means on SparkML](#).¹

Regular

Model Name. The name of the field generated after scoring to a specific cluster. Select Auto (default) or select Custom and type a name.

Number of Clusters. Specify the number of clusters to generate. The default is 5 and the minimum is 2.

Initialization

Initialization Mode. Specify the method for initializing the cluster centers. K-Means|| is the default. For details about these two methods, see [Scalable K-Means++](#).²

Initialization Steps. If the K-Means|| initialization mode is selected, specify the number of initialization steps. 2 is the default.

Advanced

Advanced Settings. Select this option if you want to set advanced options as follows.

Max Iteration. Specify the maximum number of iterations to perform when searching cluster centers. 20 is the default.

Tolerance. Specify the convergence tolerance for iterative algorithms. 1.0E-4 is the default.

Set Random Seed. Select this option and click Generate to generate the seed used by the random number generator.

Display

Display Graph. Select this option if you want a graph to be included in the output.

The following table shows the relationship between the settings in the SPSS® Modeler K-Means-AS node and the K-Means Spark parameters.

Table 1. Node properties mapped to Spark parameters

SPSS Modeler setting	Script name (property name)	K-Means SparkML parameter
Input Fields	<code>features</code>	
Number of Clusters	<code>clustersNum</code>	<code>k</code>
Initialization Mode	<code>initMode</code>	<code>initMode</code>
Initialization Steps	<code>initSteps</code>	<code>initSteps</code>
Max Iteration	<code>maxIter</code>	<code>maxIter</code>
Tolerance	<code>toleration</code>	<code>tol</code>
Random Seed	<code>randomSeed</code>	<code>seed</code>

¹ "Class KMeans." *Apache Spark*. JavaDoc. Web. 3 Oct 2017.

² Bahmani, Moseley, et al. "Scalable K-Means++." Feb 28, 2012. <http://theory.stanford.edu/%7Esergei/papers/vldb12-kmpar.pdf>.

The Cluster Viewer

Cluster models are typically used to find groups (or clusters) of similar records based on the variables examined, where the similarity between members of the same group is high and the similarity between members of different groups is low. The results can be used to identify associations that would otherwise not be apparent. For example, through cluster analysis of customer preferences, income level, and buying habits, it may be possible to identify the types of customers who are more likely to respond to a particular marketing campaign.

There are two approaches to interpreting the results in a cluster display:

- Examine clusters to determine characteristics unique to that cluster. *Does one cluster contain all the high-income borrowers? Does this cluster contain more records than the others?*
- Examine fields across clusters to determine how values are distributed among clusters. *Does one's level of education determine membership in a cluster? Does a high credit score distinguish between membership in one cluster or another?*

Using the main views and the various linked views in the Cluster Viewer, you can gain insight to help you answer these questions.

Who uses clustering?

Clustering techniques are useful in a wide variety of situations, including:

- **Market Segmentation.** Identify distinct groups among a customer base, allowing precise targeting of sales efforts.
- **Product Bundling.** Identify groups of products that tend to appeal to specific customer types.
- **Formal Classification.** Classify groups, such as plants or animals into formal taxonomies.
- **Medical Diagnosis.** Use biological patterns to uncover rules for identifying or diagnosing medical disorders.

The following cluster model nuggets can be generated in IBM® SPSS® Modeler:

- Kohonen net model nugget
- K-Means model nugget
- TwoStep cluster model nugget

To see information about the cluster model nuggets, right-click the model node and choose Browse from the context menu (or Edit for nodes in a stream). Alternatively, if you are using the Auto Cluster modeling node, double-click on the required cluster nugget within the Auto Cluster model nugget. See the topic [Auto Cluster node](#) for more information.

- [Cluster Viewer - Model Tab](#)
- [Navigating the Cluster Viewer](#)
- [Generating Graphs from Cluster Models](#)

Related information

- [Kohonen Model Nuggets](#)
 - [K-Means Model Nuggets](#)
 - [TwoStep Cluster Model Nuggets](#)
 - [Cluster Viewer - Model Tab](#)
-

Cluster Viewer - Model Tab

The Model tab for cluster models shows a graphical display of summary statistics and distributions for fields between clusters; this is known as the **Cluster Viewer**.

Note: The Model tab is not available for models built in versions of IBM® SPSS® Modeler prior to 13.

The Cluster Viewer is made up of two panels, the main view on the left and the linked, or auxiliary, view on the right. There are two main views:

- Model Summary (the default). See the topic [Model Summary View](#) for more information.
- Clusters. See the topic [Clusters View](#) for more information.

There are four linked/auxiliary views:

- Predictor Importance. See the topic [Cluster Predictor Importance View](#) for more information.
 - Cluster Sizes (the default). See the topic [Cluster Sizes View](#) for more information.
 - Cell Distribution. See the topic [Cell Distribution View](#) for more information.
 - Cluster Comparison. See the topic [Cluster Comparison View](#) for more information.
- [Model Summary View](#)
 - [Clusters View](#)
 - [Cluster Predictor Importance View](#)
 - [Cluster Sizes View](#)
 - [Cell Distribution View](#)
 - [Cluster Comparison View](#)

Related information

- [The Cluster Viewer](#)
 - [Model Summary View](#)
 - [Clusters View](#)
 - [Transpose Clusters and Features](#)
 - [Sort Features](#)
 - [Sort Clusters](#)
 - [Cell Contents](#)
 - [Cluster Predictor Importance View](#)
 - [Cluster Sizes View](#)
 - [Cell Distribution View](#)
 - [Cluster Comparison View](#)
 - [Navigating the Cluster Viewer](#)
-

Model Summary View

The Model Summary view shows a snapshot, or summary, of the cluster model, including a Silhouette measure of cluster cohesion and separation that is shaded to indicate poor, fair, or good results. This snapshot enables you to quickly check if the quality is poor, in which case you may decide to return to the modeling node to amend the cluster model settings to produce a better result.

The results of poor, fair, and good are based on the work of Kaufman and Rousseeuw (1990) regarding interpretation of cluster structures. In the Model Summary view, a good result equates to data that reflects Kaufman and Rousseeuw's rating as either reasonable or strong evidence of cluster structure, fair reflects their rating of weak evidence, and poor reflects their rating of no significant evidence.

The silhouette measure averages, over all records, $(B-A) / \max(A,B)$, where A is the record's distance to its cluster center and B is the record's distance to the nearest cluster center that it doesn't belong to. A silhouette coefficient of 1 would mean that all cases are located directly on their cluster centers. A value of -1 would mean all cases are located on the cluster centers of some other cluster. A value of 0 means, on average, cases are equidistant between their own cluster center and the nearest other cluster.

The summary includes a table that contains the following information:

- **Algorithm.** The clustering algorithm used, for example, "TwoStep".
- **Input Features.** The number of fields, also known as **inputs** or **predictors**.
- **Clusters.** The number of clusters in the solution.

Related information

- [Cluster Viewer - Model Tab](#)
 - [Clusters View](#)
 - [Transpose Clusters and Features](#)
 - [Sort Features](#)
 - [Sort Clusters](#)
 - [Cell Contents](#)
 - [Cluster Predictor Importance View](#)
 - [Cluster Sizes View](#)
 - [Cell Distribution View](#)
 - [Cluster Comparison View](#)
 - [Navigating the Cluster Viewer](#)
-

Clusters View

The Clusters view contains a cluster-by-features grid that includes cluster names, sizes, and profiles for each cluster.

The columns in the grid contain the following information:

- **Cluster.** The cluster numbers created by the algorithm.
- **Label.** Any labels applied to each cluster (this is blank by default). Double-click in the cell to enter a label that describes the cluster contents; for example, "Luxury car buyers".
- **Description.** Any description of the cluster contents (this is blank by default). Double-click in the cell to enter a description of the cluster; for example, "55+ years of age, professionals, earning over \$100,000".
- **Size.** The size of each cluster as a percentage of the overall cluster sample. Each size cell within the grid displays a vertical bar that shows the size percentage within the cluster, a size percentage in numeric format, and the cluster case counts.

- **Features.** The individual inputs or predictors, sorted by overall importance by default. If any columns have equal sizes they are shown in ascending sort order of the cluster numbers.

Overall feature importance is indicated by the color of the cell background shading; the most important feature is darkest; the least important feature is unshaded. A guide above the table indicates the importance attached to each feature cell color.

When you hover your mouse over a cell, the full name/label of the feature and the importance value for the cell is displayed. Further information may be displayed, depending on the view and feature type. In the Cluster Centers view, this includes the cell statistic and the cell value; for example: "Mean: 4.32". For categorical features the cell shows the name of the most frequent (modal) category and its percentage.

Within the Clusters view, you can select various ways to display the cluster information:

- Transpose clusters and features. See the topic [Transpose Clusters and Features](#) for more information.
- Sort features. See the topic [Sort Features](#) for more information.
- Sort clusters. See the topic [Sort Clusters](#) for more information.
- Select cell contents. See the topic [Cell Contents](#) for more information.

- [Transpose Clusters and Features](#)
- [Sort Features](#)
- [Sort Clusters](#)
- [Cell Contents](#)

Related information

- [Cluster Viewer - Model Tab](#)
- [Model Summary View](#)
- [Transpose Clusters and Features](#)
- [Sort Features](#)
- [Sort Clusters](#)
- [Cell Contents](#)
- [Cluster Predictor Importance View](#)
- [Cluster Sizes View](#)
- [Cell Distribution View](#)
- [Cluster Comparison View](#)
- [Navigating the Cluster Viewer](#)

Transpose Clusters and Features

By default, clusters are displayed as columns and features are displayed as rows. To reverse this display, click the Transpose Clusters and Features button to the left of the Sort Features By buttons. For example you may want to do this when you have many clusters displayed, to reduce the amount of horizontal scrolling required to see the data.

Related information

- [Cluster Viewer - Model Tab](#)
- [Model Summary View](#)
- [Clusters View](#)
- [Sort Features](#)
- [Sort Clusters](#)
- [Cell Contents](#)
- [Cluster Predictor Importance View](#)
- [Cluster Sizes View](#)
- [Cell Distribution View](#)
- [Cluster Comparison View](#)
- [Navigating the Cluster Viewer](#)

Sort Features

The Sort Features By buttons enable you to select how feature cells are displayed:

- **Overall Importance.** This is the default sort order. Features are sorted in descending order of overall importance, and sort order is the same across clusters. If any features have tied importance values, the tied features are listed in ascending sort order of the feature names.
- **Within-Cluster Importance.** Features are sorted with respect to their importance for each cluster. If any features have tied importance values, the tied features are listed in ascending sort order of the feature names. When this option is chosen the sort order usually varies

across clusters.

- **Name.** Features are sorted by name in alphabetical order.
- **Data order.** Features are sorted by their order in the dataset.

Related information

- [Cluster Viewer - Model Tab](#)
 - [Model Summary View](#)
 - [Clusters View](#)
 - [Transpose Clusters and Features](#)
 - [Sort Clusters](#)
 - [Cell Contents](#)
 - [Cluster Predictor Importance View](#)
 - [Cluster Sizes View](#)
 - [Cell Distribution View](#)
 - [Cluster Comparison View](#)
 - [Navigating the Cluster Viewer](#)
-

Sort Clusters

By default clusters are sorted in descending order of size. The Sort Clusters By buttons enable you to sort them by name in alphabetical order, or, if you have created unique labels, in alphanumeric label order instead.

Features that have the same label are sorted by cluster name. If clusters are sorted by label and you edit the label of a cluster, the sort order is automatically updated.

Related information

- [Cluster Viewer - Model Tab](#)
 - [Model Summary View](#)
 - [Clusters View](#)
 - [Transpose Clusters and Features](#)
 - [Sort Features](#)
 - [Cell Contents](#)
 - [Cluster Predictor Importance View](#)
 - [Cluster Sizes View](#)
 - [Cell Distribution View](#)
 - [Cluster Comparison View](#)
 - [Navigating the Cluster Viewer](#)
-

Cell Contents

The Cells buttons enable you to change the display of the cell contents for features and evaluation fields.

- **Cluster Centers.** By default, cells display feature names/labels and the central tendency for each cluster/feature combination. The mean is shown for continuous fields and the mode (most frequently occurring category) with category percentage for categorical fields.
- **Absolute Distributions.** Shows feature names/labels and absolute distributions of the features within each cluster. For categorical features, the display shows bar charts overlaid with categories ordered in ascending order of the data values. For continuous features, the display shows a smooth density plot which use the same endpoints and intervals for each cluster.
The solid red colored display shows the cluster distribution, whilst the paler display represents the overall data.
- **Relative Distributions.** Shows feature names/labels and relative distributions in the cells. In general the displays are similar to those shown for absolute distributions, except that relative distributions are displayed instead.
The solid red colored display shows the cluster distribution, while the paler display represents the overall data.
- **Basic View.** Where there are a lot of clusters, it can be difficult to see all the detail without scrolling. To reduce the amount of scrolling, select this view to change the display to a more compact version of the table.

Related information

- [Cluster Viewer - Model Tab](#)
- [Model Summary View](#)

- [Clusters View](#)
 - [Transpose Clusters and Features](#)
 - [Sort Features](#)
 - [Sort Clusters](#)
 - [Cluster Predictor Importance View](#)
 - [Cluster Sizes View](#)
 - [Cell Distribution View](#)
 - [Cluster Comparison View](#)
 - [Navigating the Cluster Viewer](#)
-

Cluster Predictor Importance View

The Predictor Importance view shows the relative importance of each field in estimating the model.

Related information

- [Cluster Viewer - Model Tab](#)
 - [Model Summary View](#)
 - [Clusters View](#)
 - [Transpose Clusters and Features](#)
 - [Sort Features](#)
 - [Sort Clusters](#)
 - [Cell Contents](#)
 - [Cluster Sizes View](#)
 - [Cell Distribution View](#)
 - [Cluster Comparison View](#)
 - [Navigating the Cluster Viewer](#)
-

Cluster Sizes View

The Cluster Sizes view shows a pie chart that contains each cluster. The percentage size of each cluster is shown on each slice; hover the mouse over each slice to display the count in that slice.

Below the chart, a table lists the following size information:

- The size of the smallest cluster (both a count and percentage of the whole).
- The size of the largest cluster (both a count and percentage of the whole).
- The ratio of size of the largest cluster to the smallest cluster.

Related information

- [Cluster Viewer - Model Tab](#)
 - [Model Summary View](#)
 - [Clusters View](#)
 - [Transpose Clusters and Features](#)
 - [Sort Features](#)
 - [Sort Clusters](#)
 - [Cell Contents](#)
 - [Cluster Predictor Importance View](#)
 - [Cell Distribution View](#)
 - [Cluster Comparison View](#)
 - [Navigating the Cluster Viewer](#)
-

Cell Distribution View

The Cell Distribution view shows an expanded, more detailed, plot of the distribution of the data for any feature cell you select in the table in the Clusters main panel.

Related information

-
- [Cluster Viewer - Model Tab](#)
 - [Model Summary View](#)
 - [Clusters View](#)
 - [Transpose Clusters and Features](#)
 - [Sort Features](#)
 - [Sort Clusters](#)
 - [Cell Contents](#)
 - [Cluster Predictor Importance View](#)
 - [Cluster Sizes View](#)
 - [Cluster Comparison View](#)
 - [Navigating the Cluster Viewer](#)
-

Cluster Comparison View

The Cluster Comparison view consists of a grid-style layout, with features in the rows and selected clusters in the columns. This view helps you to better understand the factors that make up the clusters; it also enables you to see differences between clusters not only as compared with the overall data, but with each other.

To select clusters for display, click on the top of the cluster column in the Clusters main panel. Use either Ctrl-click or Shift-click to select or deselect more than one cluster for comparison.

Note: You can select up to five clusters for display.

Clusters are shown in the order in which they were selected, while the order of fields is determined by the Sort Features By option. When you select Within-Cluster Importance, fields are always sorted by overall importance .

The background plots show the overall distributions of each features:

- Categorical features are shown as dot plots, where the size of the dot indicates the most frequent/modal category for each cluster (by feature).
- Continuous features are displayed as boxplots, which show overall medians and the interquartile ranges.

Overlaid on these background views are boxplots for selected clusters:

- For continuous features, square point markers and horizontal lines indicate the median and interquartile range for each cluster.
- Each cluster is represented by a different color, shown at the top of the view.

Related information

- [Cluster Viewer - Model Tab](#)
 - [Model Summary View](#)
 - [Clusters View](#)
 - [Transpose Clusters and Features](#)
 - [Sort Features](#)
 - [Sort Clusters](#)
 - [Cell Contents](#)
 - [Cluster Predictor Importance View](#)
 - [Cluster Sizes View](#)
 - [Cell Distribution View](#)
 - [Navigating the Cluster Viewer](#)
-

Navigating the Cluster Viewer

The Cluster Viewer is an interactive display. You can:

- Select a field or cluster to view more details.
- Compare clusters to select items of interest.
- Alter the display.
- Transpose axes.
- Generate Derive, Filter, and Select nodes using the Generate menu.

Using the Toolbars

You control the information shown in both the left and right panels by using the toolbar options. You can change the orientation of the display (top-down, left-to-right, or right-to-left) using the toolbar controls. In addition, you can also reset the viewer to the default settings, and open a dialog box to specify the contents of the Clusters view in the main panel.

The Sort Features By, Sort Clusters By, Cells, and Display options are only available when you select the Clusters view in the main panel. See the topic [Clusters View](#) for more information.

Table 1. Toolbar icons

Icon	Topic
	See Transpose Clusters and Features
	See Sort Features By
	See Sort Clusters By
	See Cells

Generating Nodes from Cluster Models

The Generate menu enables you to create new nodes based on the cluster model. This option is available from the Model tab of the generated model and enables you to generate nodes based on either the current display or selection (that is, all visible clusters or all selected ones). For example, you can select a single feature and then generate a Filter node to discard all other (nonvisible) features. The generated nodes are placed unconnected on the canvas. In addition, you can generate a copy of the model nugget to the models palette. Remember to connect the nodes and make any desired edits before execution.

- **Generate Modeling Node.** Creates a modeling node on the stream canvas. This would be useful, for example, if you have a stream in which you want to use these model settings but you no longer have the modeling node used to generate them.
- **Model to Palette.** Creates a the nugget on the Models palette. This is useful in situations where a colleague may have sent you a stream containing the model and not the model itself.
- **Filter Node.** Creates a new Filter node to filter fields that are not used by the cluster model, and/or not visible in the current Cluster Viewer display. If there is a Type node upstream from this Cluster node, any fields with the role *Target* are discarded by the generated Filter node.
- **Filter Node (from selection).** Creates a new Filter node to filter fields based on selections in the Cluster Viewer. Select multiple fields using the Ctrl-click method. Fields selected in the Cluster Viewer are discarded downstream, but you can change this behavior by editing the Filter node before execution.
- **Select Node.** Creates a new Select node to select records based on their membership in any of the clusters visible in the current Cluster Viewer display. A select condition is automatically generated.
- **Select Node (from selection).** Creates a new Select node to select records based on membership in clusters selected in the Cluster Viewer. Select multiple clusters using the Ctrl-click method.
- **Derive Node.** Creates a new Derive node, which derives a flag field that assigns records a value of *True* or *False* based on membership in all clusters visible in the Cluster Viewer. A derive condition is automatically generated.
- **Derive Node (from selection).** Creates a new Derive node, which derives a flag field based on membership in clusters selected in the Cluster Viewer. Select multiple clusters using the Ctrl-click method.

In addition to generating nodes, you can also create graphs from the Generate menu. See the topic [Generating Graphs from Cluster Models](#) for more information.

Control Cluster View Display

To control what is shown in the Clusters view on the main panel, click the Display button; the Display dialog opens.

Features. Selected by default. To hide all input features, deselect the check box.

Evaluation Fields. Choose the evaluation fields (fields not used to create the cluster model, but sent to the model viewer to evaluate the clusters) to display; none are shown by default. Note The evaluation field must be a string with more than one value. This check box is unavailable if no evaluation fields are available.

Cluster Descriptions. Selected by default. To hide all cluster description cells, deselect the check box.

Cluster Sizes. Selected by default. To hide all cluster size cells, deselect the check box.

Maximum Number of Categories. Specify the maximum number of categories to display in charts of categorical features; the default is 20.

Related information

- [Cluster Viewer - Model Tab](#)
- [Model Summary View](#)
- [Clusters View](#)
- [Transpose Clusters and Features](#)
- [Sort Features](#)
- [Sort Clusters](#)
- [Cell Contents](#)

- [Cluster Predictor Importance View](#)
 - [Cluster Sizes View](#)
 - [Cell Distribution View](#)
 - [Cluster Comparison View](#)
-

Generating Graphs from Cluster Models

Cluster models provide a lot of information; however, it may not always be in an easily accessible format for business users. To provide the data in a way that can be easily incorporated into business reports, presentations, and so on, you can produce graphs of selected data. For example, from the Cluster Viewer you can generate a graph for a selected cluster, thereby only creating a graph for the cases in that cluster.

Note: You can only generate a graph from the Cluster Viewer when the model nugget is attached to other nodes in a stream.

Generate a graph

1. Open the model nugget containing the Cluster Viewer.
2. On the Model tab select *Clusters* from the View drop-down list.
3. In the main view, select the cluster, or clusters, for which you want to produce a graph.
4. From the Generate menu, select Graph (from selection); the Graphboard Basic tab is displayed.
Note: Only the Basic and Detailed tabs are available when you display the Graphboard in this way.
5. Using either the Basic or Detailed tab settings, specify the details to be displayed on the graph.
6. Click OK to generate the graph.

The graph heading identifies the model type and cluster, or clusters, that were chosen for inclusion.

Related information

- [Cluster Viewer - Model Tab](#)
 - [Model Summary View](#)
 - [Clusters View](#)
 - [Transpose Clusters and Features](#)
 - [Sort Features](#)
 - [Sort Clusters](#)
 - [Cell Contents](#)
 - [Cluster Predictor Importance View](#)
 - [Cluster Sizes View](#)
 - [Cell Distribution View](#)
 - [Cluster Comparison View](#)
 - [Navigating the Cluster Viewer](#)
-

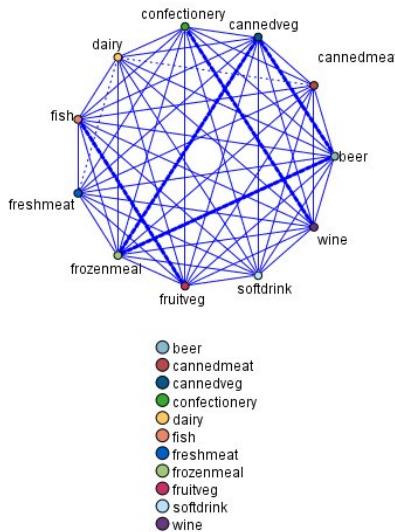
Association Rules

Association rules associate a particular conclusion (the purchase of a particular product, for example) with a set of conditions (the purchase of several other products, for example). For example, the rule

```
beer <= cannedveg & frozenmeal (173, 17.0%, 0.84)
```

states that *beer* often occurs when *cannedveg* and *frozenmeal* occur together. The rule is 84% reliable and applies to 17% of the data, or 173 records. Association rule algorithms automatically find the associations that you could find manually using visualization techniques, such as the Web node.

Figure 1. Web node showing associations between market basket items



The advantage of association rule algorithms over the more standard decision tree algorithms (C5.0 and C&R Trees) is that associations can exist between *any* of the attributes. A decision tree algorithm will build rules with only a single conclusion, whereas association algorithms attempt to find many rules, each of which may have a different conclusion.

The disadvantage of association algorithms is that they are trying to find patterns within a potentially very large search space and, hence, can require much more time to run than a decision tree algorithm. The algorithms use a **generate and test** method for finding rules--simple rules are generated initially, and these are validated against the dataset. The good rules are stored and all rules, subject to various constraints, are then specialized. **Specialization** is the process of adding conditions to a rule. These new rules are then validated against the data, and the process iteratively stores the best or most interesting rules found. The user usually supplies some limit to the possible number of antecedents to allow in a rule, and various techniques based on information theory or efficient indexing schemes are used to reduce the potentially large search space.

At the end of the processing, a table of the best rules is presented. Unlike a decision tree, this set of association rules cannot be used directly to make predictions in the way that a standard model (such as a decision tree or a neural network) can. This is due to the many different possible conclusions for the rules. Another level of transformation is required to transform the association rules into a classification rule set. Hence, the association rules produced by association algorithms are known as **unrefined models**. Although the user can browse these unrefined models, they cannot be used explicitly as classification models unless the user tells the system to generate a classification model from the unrefined model. This is done from the browser through a Generate menu option.

Two association rule algorithms are supported:

	The Apriori node extracts a set of rules from the data, pulling out the rules with the highest information content. Apriori offers five different methods of selecting rules and uses a sophisticated indexing scheme to process large data sets efficiently. For large problems, Apriori is generally faster to train; it has no arbitrary limit on the number of rules that can be retained, and it can handle rules with up to 32 preconditions. Apriori requires that input and output fields all be categorical but delivers better performance because it is optimized for this type of data.
	The Sequence node discovers association rules in sequential or time-oriented data. A sequence is a list of item sets that tends to occur in a predictable order. For example, a customer who purchases a razor and aftershave lotion may purchase shaving cream the next time he shops. The Sequence node is based on the CARMA association rules algorithm, which uses an efficient two-pass method for finding sequences.

- [Tabular versus Transactional Data](#)
- [Apriori node](#)
- [CARMA Node](#)
- [Association Rule Model Nuggets](#)
- [Sequence node](#)
- [Association Rules node](#)

Related information

- [Screening Fields and Records](#)
- [Anomaly Detection Node](#)
- [Neural Net Node](#)
- [Statistical Models](#)
- [Clustering models](#)
- [Time Series Node \(deprecated\)](#)

Tabular versus Transactional Data

Data used by association rule models may be in transactional or tabular format, as described below. These are general descriptions; specific requirements may vary as discussed in the documentation for each model type. Note that when scoring models, the data to be scored must mirror the format of the data used to build the model. Models built using tabular data can be used to score only tabular data; models built using transactional data can score only transactional data.

Transactional Format

Transactional data have a separate record for each transaction or item. If a customer makes multiple purchases, for example, each would be a separate record, with associated items linked by a customer ID. This is also sometimes known as **till-roll** format.

Customer	Purchase	
1	jam	
2	milk	
3	jam	
3	bread	
4	jam	
4	bread	
4	milk	

The Apriori, CARMA, and Sequence nodes can all use transactional data.

Tabular Data

Tabular data (also known as **basket** or **truth-table** data) have items represented by separate flags, where each flag field represents the presence or absence of a specific item. Each record represents a complete set of associated items. Flag fields can be categorical or numeric, although certain models may have more specific requirements.

Customer	Jam	Bread	Milk	
1	T	F	F	
2	F	F	T	
3	T	T	F	
4	T	T	T	

The Apriori, CARMA, GSAR, and Sequence nodes can all use tabular data.

Apriori node

The Apriori node discovers association rules in the data. Association rules are statements of the form

```
if antecedent(s) then consequent(s)
```

For example, "if a customer purchases a razor and after shave, then that customer will purchase shaving cream with 80% confidence." Apriori extracts a set of rules from the data, pulling out the rules with the highest information content. Apriori offers five different methods of selecting rules and uses a sophisticated indexing scheme to efficiently process large data sets.

Requirements. To create an Apriori rule set, you need one or more *Input* fields and one or more *Target* fields. Input and output fields (those with the role *Input*, *Target*, or *Both*) must be symbolic. Fields with the role *None* are ignored. Field types must be fully instantiated before executing the node. Data can be in tabular or transactional format. See the topic [Tabular versus Transactional Data](#) for more information.

Strengths. For large problems, Apriori is generally faster to train. It also has no arbitrary limit on the number of rules that can be retained and can handle rules with up to 32 preconditions. Apriori offers five different training methods, allowing more flexibility in matching the data mining method to the problem at hand.

- [Apriori Node Model Options](#)
- [Apriori Node Expert Options](#)

Apriori Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Minimum antecedent support. You can specify a support criterion for keeping rules in the rule set. Support refers to the percentage of records in the training data for which the antecedents (the "if" part of the rule) are true. (Note that this definition of support differs from that used in the CARMA and Sequence nodes. See the topic [Sequence Node Model Options](#) for more information.) If you are getting rules that apply to very small subsets of the data, try increasing this setting.

Note: The definition of support for Apriori is based on the number of records with the antecedents. This is in contrast to the CARMA and Sequence algorithms for which the definition of support is based on the number of records with all the items in a rule (that is, both the antecedents and consequent). The results for association models show both the (antecedent) support and rule support measures.

Minimum rule confidence. You can also specify a confidence criterion. Confidence is based on the records for which the rule's antecedents are true and is the percentage of those records for which the consequent(s) are also true. In other words, it's the percentage of predictions based on the rule that are correct. Rules with lower confidence than the specified criterion are discarded. If you are getting too many rules, try increasing this setting. If you are getting too few rules (or no rules at all), try decreasing this setting.

Note: If necessary, you can highlight the value and type in your own value. Be aware that if you reduce the confidence value below 1.0, in addition to the process requiring a lot of free memory, you might find that the rules take an extremely long time to build.

Maximum number of antecedents. You can specify the maximum number of preconditions for any rule. This is a way to limit the complexity of the rules. If the rules are too complex or too specific, try decreasing this setting. This setting also has a large influence on training time. If your rule set is taking too long to train, try reducing this setting.

Only true values for flags. If this option is selected for data in tabular (truth table) format, then only true values will be included in the resulting rules. This can help make rules easier to understand. The option does not apply to data in transactional format. See the topic [Tabular versus Transactional Data](#) for more information.

Note: The CARMA model building node ignores empty records when building a model if the field type is a flag, whereas the Apriori model building node includes empty records. Empty records are records where all the fields used in the model build have a false value.

Optimize. Select options designed to increase performance during model building based on your specific needs.

- Select Speed to instruct the algorithm to never use disk spilling in order to improve performance.
 - Select Memory to instruct the algorithm to use disk spilling when appropriate at some sacrifice to speed. This option is selected by default.
- Note: When running in distributed mode, this setting can be overridden by administrator options specified in the *options.cfg* file. For more information, see the *IBM® SPSS® Modeler Server Administrator's Guide*.

Related information

- [Apriori node](#)
 - [Apriori Node Expert Options](#)
-

Apriori Node Expert Options

For those with detailed knowledge of Apriori's operation, the following expert options allow you to fine-tune the induction process. To access expert options, set the Mode to Expert on the Expert tab.

Evaluation measure. Apriori supports five methods of evaluating potential rules.

- **Rule Confidence.** The default method uses the confidence (or accuracy) of the rule to evaluate rules. For this measure, the Evaluation measure lower bound is disabled, since it is redundant with the Minimum rule confidence option on the Model tab. See the topic [Apriori Node Model Options](#) for more information.
- **Confidence Difference.** (Also called **absolute confidence difference to prior**.) This evaluation measure is the absolute difference between the rule's confidence and its prior confidence. This option prevents bias where the outcomes are not evenly distributed. This helps prevent "obvious" rules from being kept. Set the evaluation measure lower bound to the minimum difference in confidence for which you want rules to be kept.
- **Confidence Ratio.** (Also called **difference of confidence quotient to 1**.) This evaluation measure is the ratio of rule confidence to prior confidence (or, if the ratio is greater than one, its reciprocal) subtracted from 1. Like Confidence Difference, this method takes uneven distributions into account. It is especially good at finding rules that predict rare events. Set the evaluation measure lower bound to the difference for which you want rules to be kept.
- **Information Difference.** (Also called **information difference to prior**.) This measure is based on the **information gain** measure. If the probability of a particular consequent is considered as a logical value (a **bit**), then the information gain is the proportion of that bit that can be determined, based on the antecedents. The information difference is the difference between the information gain, given the antecedents, and the information gain, given only the prior confidence of the consequent. An important feature of this method is that it takes support into account so that rules that cover more records are preferred for a given level of confidence. Set the evaluation measure lower bound to the information difference for which you want rules to be kept.
- **Normalized Chi-square.** (Also called **normalized chi-squared measure**.) This measure is a statistical index of association between antecedents and consequents. The measure is normalized to take values between 0 and 1. This measure is even more strongly dependent on support than the information difference measure. Set the evaluation measure lower bound to the information difference for which you want rules to be kept.

Allow rules without antecedents. Select to allow rules that include only the consequent (item or item set). This is useful when you are interested in determining common items or item sets. For example, `cannedveg` is a single-item rule without an antecedent that indicates purchasing `cannedveg` is a common occurrence in the data. In some cases, you may want to include such rules if you are simply interested in the most confident predictions. This option is off by default. By convention, antecedent support for rules without antecedents is expressed as 100%, and rule support will be the same as confidence.

Related information

- [Apriori node](#)
 - [Apriori Node Model Options](#)
-

CARMA Node

The CARMA node uses an association rules discovery algorithm to discover association rules in the data. Association rules are statements in the form

```
if antecedent(s) then consequent(s)
```

For example, if a Web customer purchases a wireless card and a high-end wireless router, the customer is also likely to purchase a wireless music server if offered. The CARMA model extracts a set of rules from the data without requiring you to specify input or target fields. This means that the rules generated can be used for a wider variety of applications. For example, you can use rules generated by this node to find a list of products or services (antecedents) whose consequent is the item that you want to promote this holiday season. Using IBM® SPSS® Modeler, you can determine which clients have purchased the antecedent products and construct a marketing campaign designed to promote the consequent product.

Requirements. In contrast to Apriori, the CARMA node does not require *Input* or *Target* fields. This is integral to the way the algorithm works and is equivalent to building an Apriori model with all fields set to *Both*. You can constrain which items are listed only as antecedents or consequents by filtering the model after it is built. For example, you can use the model browser to find a list of products or services (antecedents) whose consequent is the item that you want to promote this holiday season.

To create a CARMA rule set, you need to specify an ID field and one or more content fields. The ID field can have any role or measurement level. Fields with the role *None* are ignored. Field types must be fully instantiated before executing the node. Like Apriori, data may be in tabular or transactional format. See the topic [Tabular versus Transactional Data](#) for more information.

Strengths. The CARMA node is based on the CARMA association rules algorithm. In contrast to Apriori, the CARMA node offers build settings for rule support (support for both antecedent and consequent) rather than antecedent support. CARMA also allows rules with multiple consequents. Like Apriori, models generated by a CARMA node can be inserted into a data stream to create predictions. See the topic [Model Nuggets](#) for more information.

- [CARMA Node Fields Options](#)
- [CARMA Node Model Options](#)
- [CARMA Node Expert Options](#)

Related information

- [Overview of modeling nodes](#)
 - [CARMA Node Fields Options](#)
 - [CARMA Node Model Options](#)
 - [CARMA Node Expert Options](#)
-

CARMA Node Fields Options

Before executing a CARMA node, you must specify input fields on the Fields tab of the CARMA node. While most modeling nodes share identical Fields tab options, the CARMA node contains several unique options. All options are discussed below.

Use Type node settings. This option tells the node to use field information from an upstream type node. This is the default.

Use custom settings. This option tells the node to use field information specified here instead of that given in any upstream Type node(s). After selecting this option, specify fields below according to whether you are reading data in transactional or tabular format.

Use transactional format. This option changes the field controls in the rest of this dialog box depending on whether your data are in transactional or tabular format. If you use multiple fields with transactional data, the items specified in these fields for a particular record are assumed to represent items found in a single transaction with a single timestamp. See the topic [Tabular versus Transactional Data](#) for more information.

Tabular data

If Use transactional format is not selected, the following fields are displayed.

- Inputs. Select the input field or fields. This is similar to setting the field role to *Input* in a Type node.
- Partition. This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to generate the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)

Transactional data

If you select Use transactional format, the following fields are displayed.

- ID. For transactional data, select an ID field from the list. Numeric or symbolic fields can be used as the ID field. Each unique value of this field should indicate a specific unit of analysis. For example, in a market basket application, each ID might represent a single customer. For a Web log analysis application, each ID might represent a computer (by IP address) or a user (by login data).
- IDs are contiguous. (Apriori and CARMA nodes only) If your data are presorted so that all records with the same ID are grouped together in the data stream, select this option to speed up processing. If your data are not presorted (or you are not sure), leave this option unselected and the node will sort the data automatically.
Note: If your data are not sorted and you select this option, you may get invalid results in your model.
- Content. Specify the content field(s) for the model. These fields contain the items of interest in association modeling. You can specify multiple flag fields (if data are in tabular format) or a single nominal field (if data are in transactional format).

Related information

- [CARMA Node](#)
 - [CARMA Node Model Options](#)
 - [CARMA Node Expert Options](#)
-

CARMA Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Minimum rule support (%). You can specify a support criterion. Rule support refers to the proportion of IDs in the training data that contain the entire rule. (Note that this definition of support differs from antecedent support used in the Apriori nodes.) If you want to focus on more common rules, increase this setting.

Minimum rule confidence (%). You can specify a confidence criterion for keeping rules in the rule set. Confidence refers to the percentage of IDs where a correct prediction is made (out of all IDs for which the rule makes a prediction). It is calculated as the number of IDs for which the entire rule is found divided by the number of IDs for which the antecedents are found, based on the training data. Rules with lower confidence than the specified criterion are discarded. If you are getting uninteresting or too many rules, try increasing this setting. If you are getting too few rules, try decreasing this setting.

Note: If necessary, you can highlight the value and type in your own value. Be aware that if you reduce the confidence value below 1.0, in addition to the process requiring a lot of free memory, you might find that the rules take an extremely long time to build.

Maximum rule size. You can set the maximum number of distinct **item sets** (as opposed to **items**) in a rule. If the rules of interest are relatively short, you can decrease this setting to speed up building the rule set.

Note: The CARMA model building node ignores empty records when building a model if the field type is a flag, whereas the Apriori model building node includes empty records. Empty records are records where all the fields used in the model build have a false value.

Related information

- [CARMA Node](#)
 - [CARMA Node Fields Options](#)
 - [CARMA Node Expert Options](#)
-

CARMA Node Expert Options

For those with detailed knowledge of the CARMA node's operation, the following expert options allow you to fine-tune the model-building process. To access expert options, set the mode to Expert on the Expert tab.

Exclude rules with multiple consequents. Select to exclude "two-headed" consequents—that is, consequents that contain two items. For example, the rule `bread & cheese & fish`

-> `wine&fruit` contains a two-headed consequent, `wine&fruit`. By default, such rules are included.

Set pruning value. To conserve memory, the CARMA algorithm used periodically removes (**prunes**) infrequent item sets from its list of potential item sets during processing. Select this option to adjust the frequency of pruning, and the number you specify determines the frequency of pruning. Enter a smaller value to decrease the memory requirements of the algorithm (but potentially increase the training time required), or enter a larger value to speed up training (but potentially increase memory requirements). The default value is 500.

Vary support. Select to increase efficiency by excluding infrequent item sets that seem to be frequent when they are included unevenly. This is achieved by starting with a higher support level and tapering it down to the level specified on the Model tab. Enter a value for Estimated number of transactions to specify how quickly the support level should be tapered.

Allow rules without antecedents. Select to allow rules that include only the consequent (item or item set). This is useful when you are interested in determining common items or item sets. For example, `cannedveg` is a single-item rule without an antecedent that indicates purchasing `cannedveg` is a common occurrence in the data. In some cases, you may want to include such rules if you are simply interested in the most confident predictions. This option is unselected by default.

Related information

- [CARMA Node](#)
 - [CARMA Node Fields Options](#)
 - [CARMA Node Model Options](#)
-

Association Rule Model Nuggets

Association rule model nuggets represent the rules discovered by one of the following association rule modeling nodes:

- Apriori
- CARMA

The model nuggets contain information about the rules extracted from the data during model building.

Note: Association rule nugget scoring may be incorrect if you do not sort transactional data by ID.
Viewing Results

You can browse the rules generated by association models (Apriori and CARMA) and Sequence models using the Model tab on the dialog box. Browsing a model nugget shows you the information about the rules and provides options for filtering and sorting results before generating new nodes or scoring the model. See the topic [Association Rule model nugget details](#) for more information.

Scoring the Model

Refined model nuggets (Apriori, CARMA, and Sequence) may be added to a stream and used for scoring. See the topic [Using Model Nuggets in Streams](#) for more information. Model nuggets used for scoring include an extra Settings tab on their respective dialog boxes. See the topic [Association Rule Model Nugget Settings](#) for more information.

An unrefined model nugget cannot be used for scoring in its raw format. Instead, you can generate a rule set and use the rule set for scoring. See the topic [Generating a Rule Set from an Association Model Nugget](#) for more information.

- [Association Rule model nugget details](#)
- [Association Rule Model Nugget Settings](#)
- [Association Rule Model Nugget Summary](#)
- [Generating a Rule Set from an Association Model Nugget](#)
- [Generating a Filtered Model](#)
- [Scoring Association Rules](#)
- [Deploying Association Models](#)

Related information

- [Association Rule model nugget details](#)
- [Specifying Filters for Rules](#)
- [Generating Graphs for Rules](#)
- [Association Rule Model Nugget Summary](#)
- [Generating a Rule Set from an Association Model Nugget](#)

- [Generating a Filtered Model](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)
- [Apriori node](#)

Association Rule model nugget details

On the Model tab of an Association Rule model nugget, you can see a table containing the rules extracted by the algorithm. Each row in the table represents a rule. The first column represents the consequents (the "then" part of the rule), while the next column represents the antecedents (the "if" part of the rule). Subsequent columns contain rule information, such as confidence, support, and lift.

Association rules are often shown in the format in the following table.

Table 1. Example of an association rule	
Consequent	Antecedent
Drug = drugY	Sex = F BP = HIGH

The example rule is interpreted as *if Sex = "F" and BP = "HIGH," then Drug is likely to be drugY*; or to phrase it another way, *for records where Sex = "F" and BP = "HIGH," Drug is likely to be drugY*. Using the dialog box toolbar, you can choose to display additional information, such as confidence, support, and instances.

Sort menu. The Sort menu button on the toolbar controls the sorting of rules. Direction of sorting (ascending or descending) can be changed using the sort direction button (up or down arrow).

You can sort rules by:

- Support
- Confidence
- Rule Support
- Consequent
- Evaluation
- Lift
- Deployability

Show/Hide menu. The Show/Hide menu (criteria toolbar button) controls options for the display of rules.

Figure 1. Show/Hide button



The following display options are available:

- Rule ID displays the rule ID assigned during model building. A rule ID enables you to identify which rules are being applied for a given prediction. Rule IDs also allow you to merge additional rule information, such as deployability, product information, or antecedents, at a later time.
- Instances displays information about the number of unique IDs to which the rule applies—that is, for which the antecedents are true. For example, given the rule `bread -> cheese`, the number of records in the training data that include the antecedent `bread` are referred to as **instances**.
- Support displays antecedent support—that is, the proportion of IDs for which the antecedents are true, based on the training data. For example, if 50% of the training data includes the purchase of bread, then the rule `bread -> cheese` will have an antecedent support of 50%. Note: Support as defined here is the same as the instances but is represented as a percentage.
- Confidence displays the ratio of rule support to antecedent support. This indicates the proportion of IDs with the specified antecedent(s) for which the consequent(s) is/are also true. For example, if 50% of the training data contains bread (indicating antecedent support) but only 20% contains both bread and cheese (indicating rule support), then confidence for the rule `bread -> cheese` would be **Rule Support / Antecedent Support** or, in this case, 40%.
- Rule Support displays the proportion of IDs for which the entire rule, antecedents, and consequent(s), are true. For example, if 20% of the training data contains both bread and cheese, then rule support for the rule `bread -> cheese` is 20%.
- Evaluation is included if you select one of the expert association rule criterion (Confidence Difference, Confidence Ratio, Information Difference, or Normalized Chi-Square). These expert criteria measures are compared to the Evaluation measure lower bound number set by the user (and only applies when an expert criterion rule is selected). The Evaluation statistic has the following meanings for each expert association rule criterion:
 - Confidence Difference: Posterior Confidence - Prior Confidence
 - Confidence Ratio: (Posterior Confidence - Prior Confidence)/Posterior Confidence

- Information Difference: Information Gain Measure
- Normalized Chi-Square: Normalized Chi-Square Statistic

Each of these statistics are compared to the Evaluation measure lower bound number set by the user, and a rule is selected if the statistics exceed this number.

- Lift displays the ratio of confidence for the rule to the prior probability of having the consequent. For example, if 10% of the entire population buys bread, then a rule that predicts whether people will buy bread with 20% confidence will have a lift of $20/10 = 2$. If another rule tells you that people will buy bread with 11% confidence, then the rule has a lift of close to 1, meaning that having the antecedent(s) does not make a lot of difference in the probability of having the consequent. In general, rules with lift different from 1 will be more interesting than rules with lift close to 1.
- Deployability is a measure of what percentage of the training data satisfies the conditions of the antecedent but does not satisfy the consequent. In product purchase terms, it basically means what percentage of the total customer base owns (or has purchased) the antecedent(s) but has not yet purchased the consequent. The deployability statistic is defined as $((\text{Antecedent Support in } \# \text{ of Records} - \text{Rule Support in } \# \text{ of Records}) / \text{Number of Records}) * 100$, where *Antecedent Support* means the number of records for which the antecedents are true and *Rule Support* means the number of records for which both antecedents and the consequent are true.

Filter button. The Filter button (funnel icon) on the menu expands the bottom of the dialog box to show a panel where active rule filters are displayed. Filters are used to narrow the number of rules displayed on the Models tab.

Figure 2. Filter button



To create a filter, click the Filter icon to the right of the expanded panel. This opens a separate dialog box in which you can specify constraints for displaying rules. Note that the Filter button is often used in conjunction with the Generate menu to first filter rules and then generate a model containing that subset of rules. For more information, see [Specifying Filters for Rules](#) below.

Find Rule button. The Find Rule button (binoculars icon) enables you to search the rules shown for a specified rule ID. The adjacent display box indicates the number of rules currently displayed out of the number available. Rule IDs are assigned by the model in the order of discovery at the time and are added to the data during scoring.

Figure 3. Find Rule button



To reorder rule IDs:

1. You can rearrange rule IDs in IBM® SPSS® Modeler by first sorting the rule display table according to the desired measurement, such as confidence or lift.
2. Then using options from the Generate menu, create a filtered model.
3. In the Filtered Model dialog box, select Renumber rules consecutively starting with, and specify a start number.

See [Generating a Filtered Model](#) for more information.

- [Specifying Filters for Rules](#)
- [Generating Graphs for Rules](#)

Specifying Filters for Rules

By default, rule algorithms, such as Apriori, CARMA, and Sequence, may generate a large and cumbersome number of rules. To enhance clarity when browsing or to streamline rule scoring, you should consider filtering rules so that consequents and antecedents of interest are more prominently displayed. Using the filtering options on the Model tab of a rule browser, you can open a dialog box for specifying filter qualifications.

To create a filter, click the Edit Filters button (funnel icon) to the right of the expanded panel. This opens a separate Edit Filters dialog box where you can specify constraints for displaying rules.

Consequents. Select Enable Filter to activate options for filtering rules based on the inclusion or exclusion of specified consequents. Select Includes any of to create a filter where rules contain at least one of the specified consequents. Alternatively, select Excludes to create a filter excluding specified consequents. You can select consequents using the picker icon to the right of the list box. This opens a dialog box listing all consequents present in the generated rules.

Note: Consequents may contain more than one item. Filters will check only that a consequent contains one of the items specified.

Antecedents. Select Enable Filter to activate options for filtering rules based on the inclusion or exclusion of specified antecedents. You can select items using the picker icon to the right of the list box. This opens a dialog box listing all antecedents present in the generated rules.

- Select Includes all of to set the filter as an inclusionary one where all antecedents specified must be included in a rule.
- Select Includes any of to create a filter where rules contain at least one of the specified antecedents.
- Select Excludes to create a filter excluding rules that contain a specified antecedent.

Confidence. Select Enable Filter to activate options for filtering rules based on the level of confidence for a rule. You can use the Min and Max controls to specify a confidence range. When you are browsing generated models, confidence is listed as a percentage. When you are scoring output, confidence is expressed as a number between 0 and 1.

Antecedent Support. Select Enable Filter to activate options for filtering rules based on the level of antecedent support for a rule. Antecedent support indicates the proportion of training data that contains the same antecedents as the current rule, making it analogous to a popularity index. You can use the Min and Max controls to specify a range used to filter rules based on support level.

Lift. Select Enable Filter to activate options for filtering rules based on the lift measurement for a rule. *Note:* Lift filtering is available only for association models built after release 8.5 or for earlier models that contain a lift measurement. Sequence models do not contain this option.

Click OK to apply all filters that have been enabled in this dialog box.

Related information

- [Association Rule Model Nuggets](#)
 - [Association Rule model nugget details](#)
 - [Generating Graphs for Rules](#)
 - [Association Rule Model Nugget Summary](#)
 - [Generating a Rule Set from an Association Model Nugget](#)
 - [Generating a Filtered Model](#)
-

Generating Graphs for Rules

The Association nodes provide a lot of information; however, it may not always be in an easily accessible format for business users. To provide the data in a way that can be easily incorporated into business reports, presentations, and so on, you can produce graphs of selected data. From the Model tab, you can generate a graph for a selected rule, thereby only creating a graph for the cases in that rule.

1. On the Model tab, select the rule in which you are interested.
2. From the Generate menu, select Graph (from selection). The Graphboard Basic tab is displayed.
Note: Only the Basic and Detailed tabs are available when you display the Graphboard in this way.
3. Using either the Basic or Detailed tab settings, specify the details to be displayed on the graph.
4. Click OK to generate the graph.

The graph heading identifies the rule and antecedent details that were chosen for inclusion.

Related information

- [Association Rule Model Nuggets](#)
 - [Association Rule model nugget details](#)
 - [Specifying Filters for Rules](#)
 - [Association Rule Model Nugget Summary](#)
 - [Generating a Rule Set from an Association Model Nugget](#)
 - [Generating a Filtered Model](#)
-

Association Rule Model Nugget Settings

This Settings tab is used to specify scoring options for association models (Apriori and CARMA). This tab is available only after the model nugget has been added to a stream for purposes of scoring.

Note: The dialog box for browsing an unrefined model does not include the Settings tab, since it cannot be scored. To score the "unrefined" model, you must first generate a rule set. See the topic [Generating a Rule Set from an Association Model Nugget](#) for more information.

Maximum number of predictions Specify the maximum number of predictions included for each set of basket items. This option is used in conjunction with Rule Criterion below to produce the "top" predictions, where *top* indicates the highest level of confidence, support, lift, and so on, as specified below.

Rule Criterion Select the measure used to determine the strength of rules. Rules are sorted by the strength of criteria selected here in order to return the top predictions for an item set. Available criteria are shown in the following list.

- Confidence
- Support
- Rule support (Support * Confidence)
- Lift

- Deployability

Allow repeat predictions Select to include multiple rules with the same consequent when scoring. For example, selecting this option allows the following rules to be scored:

```
bread & cheese -> wine
cheese & fruit -> wine
```

Turn off this option to exclude repeat predictions when scoring.

Note: Rules with multiple consequents (**bread & cheese & fruit -> wine & pate**) are considered repeat predictions only if all consequents (**wine & pate**) have been predicted before.

Ignore unmatched basket items Select to ignore the presence of additional items in the item set. For example, when this option is selected for a basket that contains [**tent & sleeping bag & kettle**], the rule **tent & sleeping bag -> gas_stove** will apply despite the extra item (**kettle**) present in the basket.

There may be some circumstances where extra items should be excluded. For example, it is likely that someone who purchases a tent, sleeping bag, and kettle may already have a gas stove, indicated by the presence of the kettle. In other words, a gas stove may not be the best prediction. In such cases, you should deselect Ignore unmatched basket items to ensure that rule antecedents exactly match the contents of a basket. By default, unmatched items are ignored.

Check that predictions are not in basket. Select to ensure that consequents are not also present in the basket. For example, if the purpose of scoring is to make a home furniture product recommendation, then it is unlikely that a basket that already contains a dining room table will be likely to purchase another one. In such a case, you should select this option. On the other hand, if products are perishable or disposable (such as cheese, baby formula, or tissue), then rules where the consequent is already present in the basket may be of value. In the latter case, the most useful option might be Do not check basket for predictions below.

Check that predictions are in basket Select this option to ensure that consequents are also present in the basket. This approach is useful when you are attempting to gain insight into existing customers or transactions. For example, you may want to identify rules with the highest lift and then explore which customers fit these rules.

Do not check basket for predictions Select to include all rules when scoring, regardless of the presence or absence of consequents in the basket.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Related information

- [Decision Tree/Rule Set model nugget settings](#)
- [C&R Tree, CHAID, QUEST, C5.0, and Apriori rule set model nuggets](#)
- [Rule Set Model Tab](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)

Association Rule Model Nugget Summary

The Summary tab of an association rule model nugget displays the number of rules discovered and the minimum and maximum for support, lift, confidence and deployability of rules in the rule set.

Related information

- [Association Rule Model Nuggets](#)
- [Association Rule model nugget details](#)
- [Specifying Filters for Rules](#)
- [Generating Graphs for Rules](#)
- [Generating a Rule Set from an Association Model Nugget](#)
- [Generating a Filtered Model](#)

- [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Generating a Rule Set from an Association Model Nugget

Association model nuggets, such as Apriori and CARMA, can be used to score data directly, or you can first generate a subset of rules, known as a **rule set**. Rule sets are particularly useful when you are working with an unrefined model, which cannot be used directly for scoring. See the topic [Unrefined Models](#) for more information.

To generate a rule set, choose Rule set from the Generate menu in the model nugget browser. You can specify the following options for translating the rules into a rule set:

Rule set name. Allows you to specify the name of the new generated Rule Set node.

Create node on. Controls the location of the new generated Rule Set node. Select Canvas, GM Palette, or Both.

Target field. Determines which output field will be used for the generated Rule Set node. Select a single output field from the list.

Minimum support. Specify the minimum support for rules to be preserved in the generated rule set. Rules with support less than the specified value will not be included in the new rule set.

Minimum confidence. Specify the minimum confidence for rules to be preserved in the generated rule set. Rules with confidence less than the specified value will not be included in the new rule set.

Default value. Allows you to specify a default value for the target field that is assigned to scored records for which no rule fires.

Related information

- [Association Rule Model Nuggets](#)
 - [Association Rule model nugget details](#)
 - [Specifying Filters for Rules](#)
 - [Generating Graphs for Rules](#)
 - [Association Rule Model Nugget Summary](#)
 - [Generating a Filtered Model](#)
 - [Sequence Model Nuggets](#)
 - [Sequence Model Nugget Details](#)
 - [Sequence Model Nugget Settings](#)
 - [Sequence Model Nugget Summary](#)
 - [Generating a Rule SuperNode from a Sequence Model Nugget](#)
-

Generating a Filtered Model

To generate a filtered model from an association model nugget, such as an Apriori, CARMA, or Sequence Rule Set node, choose Filtered Model from the Generate menu in the model nugget browser. This creates a subset model that includes only those rules currently displayed in the browser. *Note:* You cannot generate filtered models for unrefined models.

You can specify the following options for filtering rules:

Name for New Model. Allows you to specify the name of the new Filtered Model node.

Create node on. Controls the location of the new Filtered Model node. Select Canvas, GM Palette, or Both.

Rule numbering. Specify how rule IDs will be numbered in the subset of rules included in the filtered model.

- **Retain original rule ID numbers.** Select to maintain the original numbering of rules. By default, rules are given an ID that corresponds with their order of discovery by the algorithm. That order may vary depending on the algorithm employed.
- **Renumber rules consecutively starting with.** Select to assign new rule IDs for the filtered rules. New IDs are assigned based on the sort order displayed in the rule browser table on the Model tab, beginning with the number you specify here. You can specify the start number for IDs using the arrows to the right.

Related information

- [Association Rule Model Nuggets](#)

- [Association Rule model nugget details](#)
- [Specifying Filters for Rules](#)
- [Generating Graphs for Rules](#)
- [Association Rule Model Nugget Summary](#)
- [Generating a Rule Set from an Association Model Nugget](#)
- [Sequence Model Nuggets](#)
- [Sequence Model Nugget Details](#)
- [Sequence Model Nugget Settings](#)
- [Sequence Model Nugget Summary](#)
- [Generating a Rule SuperNode from a Sequence Model Nugget](#)

Scoring Association Rules

Scores produced by running new data through an association rule model nugget are returned in separate fields. Three new fields are added for each prediction, with *P* representing the prediction, *C* representing confidence, and *I* representing the rule ID. The organization of these output fields depends on whether the input data are in transactional or tabular format. See [Tabular versus Transactional Data](#) for an overview of these formats.

For example, suppose you are scoring basket data using a model that generates predictions based on the following three rules:

```
Rule_15 bread&wine -> meat (confidence 54%)
Rule_22 cheese -> fruit (confidence 43%)
Rule_5 bread&cheese -> frozveg (confidence 24%)
```

Tabular data. For tabular data, the three predictions (3 is the default) are returned in a single record.

Table 1. Scores in tabular format

ID	Bread	Wine	Cheese	P1	C1	I1	P2	C2	I2	P3	C3	I3
Fred	1	1	1	meat	0.54	15	fruit	0.43	22	frozveg	.24	5

Transactional data. For transactional data, a separate record is generated for each prediction. Predictions are still added in separate columns, but scores are returned as they are calculated. This results in records with incomplete predictions, as shown in the sample output below. The second and third predictions (P2 and P3) are blank in the first record, along with the associated confidences and rule IDs. As scores are returned, however, the final record contains all three predictions.

Table 2. Scores in transactional format

ID	Item	P1	C1	I1	P2	C2	I2	P3	C3	I3
Fred	bread	meat	0.54	14	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$
Fred	cheese	meat	0.54	14	fruit	0.43	22	\$null\$	\$null\$	\$null\$
Fred	wine	meat	0.54	14	fruit	0.43	22	frozveg	0.24	5

To include only complete predictions for reporting or deployment purposes, use a Select node to select complete records.

Note: The field names used in these examples are abbreviated for clarity. During actual use, results fields for association models are named as shown in the following table.

Table 3. Names of results fields for association models

New field	Example field name
Prediction	\$A-TRANSACTION_NUMBER-1
Confidence (or other criterion)	\$AC-TRANSACTION_NUMBER-1
Rule ID	\$A-Rule_ID-1

Rules with Multiple Consequents

The CARMA algorithm allows rules with multiple consequents—for example:

```
bread -> wine&cheese
```

When you are scoring such “two-headed” rules, predictions are returned in the format displayed in the following table.

Table 4. Scoring results including a prediction with multiple consequents

ID	Bread	Wine	Cheese	P1	C1	I1	P2	C2	I2	P3	C3	I3
Fred	1	1	1	meat&veg	0.54	16	fruit	0.43	22	frozveg	.24	5

In some cases, you may need to split such scores before deployment. To split a prediction with multiple consequents, you will need to parse the field using the CLEM string functions.

Deploying Association Models

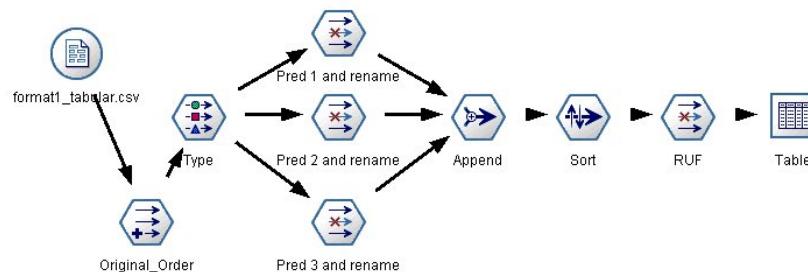
When scoring association models, predictions and confidences are output in separate columns (where P represents the prediction, C represents confidence, and I represents the rule ID). This is the case whether the input data are tabular or transactional. See the topic [Scoring Association Rules](#) for more information.

When preparing scores for deployment, you might find that your application requires you to transpose your output data to a format with predictions in rows rather than columns (one prediction per row, sometimes known as "till-roll" format).

Transposing Tabular Scores

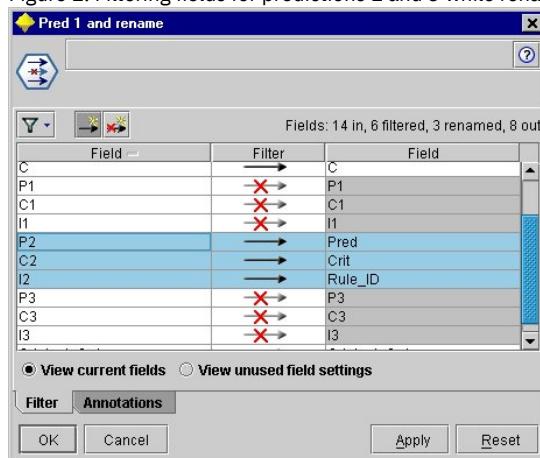
You can transpose tabular scores from columns to rows using a combination of steps in IBM® SPSS® Modeler, as described in the steps that follow.

Figure 1. Example stream used to transpose tabular data into till-roll format



1. Use the `@INDEX` function in a Derive node to ascertain the current order of predictions and save this indicator in a new field, such as `Original_order`.
2. Add a Type node to ensure that all fields are instantiated.
3. Use a Filter node to rename the default prediction, confidence, and ID fields ($P1, C1, I1$) to common fields, such as `Pred`, `Crit`, and `Rule_ID`, which will be used to append records later on. You will need one Filter node for each prediction generated.

Figure 2. Filtering fields for predictions 1 and 3 while renaming fields for prediction 2.



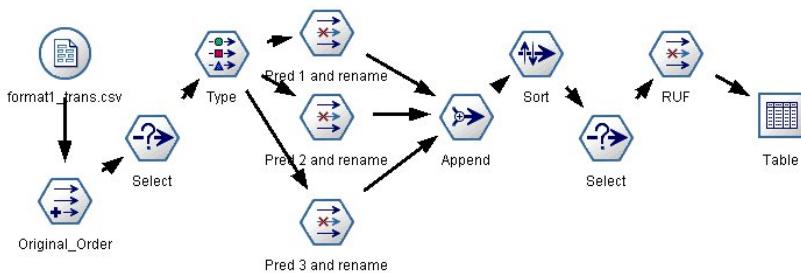
4. Use an Append node to append values for the shared `Pred`, `Crit`, and `Rule_ID`.
5. Attach a Sort node to sort records in ascending order for the field `Original_order` and in descending order for `Crit`, which is the field used to sort predictions by criteria such as confidence, lift, and support.
6. Use another Filter node to filter the field `Original_order` from the output.

At this point, the data are ready for deployment.

Transposing Transactional Scores

The process is similar for transposing transactional scores. For example, the stream shown below transposes scores to a format with a single prediction in each row as needed for deployment.

Figure 3. Example stream used to transpose transactional data into till-roll format



With the addition of two Select nodes, the process is identical to that explained earlier for tabular data.

- The first Select node is used to compare rule IDs across adjacent records and include only unique or undefined records. This Select node uses the CLEM expression to select records: `ID /= @OFFSET(ID,-1) or @OFFSET(ID,-1) = undef`.
- The second Select node is used to discard extraneous rules, or rules where `Rule_ID` has a null value. This Select node uses the following CLEM expression to discard records: `not (@NULL(Rule_ID))`.

For more information on transposing scores for deployment, contact Technical Support.

Sequence node

The Sequence node discovers patterns in sequential or time-oriented data, in the format `bread -> cheese`. The elements of a sequence are **item sets** that constitute a single transaction. For example, if a person goes to the store and purchases bread and milk and then a few days later returns to the store and purchases some cheese, that person's buying activity can be represented as two item sets. The first item set contains bread and milk, and the second one contains cheese. A **sequence** is a list of item sets that tend to occur in a predictable order. The Sequence node detects frequent sequences and creates a generated model node that can be used to make predictions.

Requirements. To create a Sequence rule set, you need to specify an ID field, an optional time field, and one or more content fields. Note that these settings must be made on the Fields tab of the modeling node; they cannot be read from an upstream Type node. The ID field can have any role or measurement level. If you specify a time field, it can have any role but its storage must be numeric, date, time, or timestamp. If you do not specify a time field, the Sequence node will use an implied timestamp, in effect using row numbers as time values. Content fields can have any measurement level and role, but all content fields must be of the same type. If they are numeric, they must be integer ranges (not real ranges).

Strengths. The Sequence node is based on the CARMA association rules algorithm, which uses an efficient two-pass method for finding sequences. In addition, the generated model node created by a Sequence node can be inserted into a data stream to create predictions. The generated model node can also generate SuperNodes for detecting and counting specific sequences and for making predictions based on specific sequences.

- [Sequence Node Fields Options](#)
- [Sequence Node Model Options](#)
- [Sequence Node Expert Options](#)
- [Sequence Model Nuggets](#)

Sequence Node Fields Options

Before executing a Sequence node, you must specify ID and content fields on the Fields tab of the Sequence node. If you want to use a time field, you also need to specify that here.

ID field. Select an ID field from the list. Numeric or symbolic fields can be used as the ID field. Each unique value of this field should indicate a specific unit of analysis. For example, in a market basket application, each ID might represent a single customer. For a Web log analysis application, each ID might represent a computer (by IP address) or a user (by login data).

- IDs are contiguous.** If your data are presorted so that all records with the same ID are grouped together in the data stream, select this option to speed up processing. If your data are not presorted (or you are not sure), leave this option unselected, and the Sequence node will sort the data automatically.

Note: If your data are not sorted and you select this option, you may get invalid results in your Sequence model.

Time field. If you want to use a field in the data to indicate event times, select Use time field and specify the field to be used. The time field must be numeric, date, time, or timestamp. If no time field is specified, records are assumed to arrive from the data source in sequential order, and record numbers are used as time values (the first record occurs at time "1"; the second, at time "2", and so on).

Content fields. Specify the content field(s) for the model. These fields contain the events of interest in sequence modeling.

The Sequence node can handle data in either tabular or transactional format. If you use multiple fields with transactional data, the items specified in these fields for a particular record are assumed to represent items found in a single transaction with a single timestamp. See the topic [Tabular versus Transactional Data](#) for more information.

Partition. This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to generate the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)

Related information

- [Sequence node](#)
 - [Sequence Node Model Options](#)
 - [Sequence Node Expert Options](#)
-

Sequence Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Minimum rule support (%) You can specify a support criterion. *Rule support* refers to the proportion of IDs in the training data that contain the entire sequence. If you want to focus on more common sequences, increase this setting.

Minimum rule confidence (%) You can specify a confidence criterion for keeping sequences in the sequence set. *Confidence* refers to the percentage of the IDs where a correct prediction is made, out of all the IDs for which the rule makes a prediction. It is calculated as the number of IDs for which the entire sequence is found divided by the number of IDs for which the antecedents are found, based on the training data. Sequences with lower confidence than the specified criterion are discarded. If you are getting too many sequences or uninteresting sequences, try increasing this setting. If you are getting too few sequences, try decreasing this setting.

Note: If necessary, you can highlight the value and type in your own value. Be aware that if you reduce the confidence value below 1.0, in addition to the process requiring a lot of free memory, you might find that the rules take an extremely long time to build.

Maximum sequence size You can set the maximum number of distinct items in a sequence. If the sequences of interest are relatively short, you can decrease this setting to speed up building the sequence set.

Predictions to add to stream Specify the number of predictions to be added to the stream by the resulting generated Model node. For more information, see [Sequence Model Nuggets](#).

Related information

- [Sequence node](#)
 - [Sequence Node Fields Options](#)
 - [Sequence Node Expert Options](#)
-

Sequence Node Expert Options

For those with detailed knowledge of the Sequence node's operation, the following expert options allow you to fine-tune the model-building process. To access expert options, set the Mode to Expert on the Expert tab.

Set maximum duration. If this option is selected, sequences will be limited to those with a duration (the time between the first and last item set) less than or equal to the value specified. If you haven't specified a time field, the duration is expressed in terms of rows (records) in the raw data. If the time field used is a time, date, or timestamp field, the duration is expressed in seconds. For numeric fields, the duration is expressed in the same units as the field itself.

Set pruning value. The CARMA algorithm used in the Sequence node periodically removes (**prunes**) infrequent item sets from its list of potential item sets during processing to conserve memory. Select this option to adjust the frequency of pruning. The number specified determines the frequency of pruning. Enter a smaller value to decrease the memory requirements of the algorithm (but potentially increase the training time required), or enter a larger value to speed up training (but potentially increase memory requirements).

Set maximum sequences in memory. If this option is selected, the CARMA algorithm will limit its memory store of candidate sequences during model building to the number of sequences specified. Select this option if IBM® SPSS® Modeler is using too much memory during the building of

Sequence models. Note that the maximum sequences value you specify here is the number of candidate sequences tracked internally as the model is built. This number should be much larger than the number of sequences you expect in the final model.

Constrain gaps between item sets. This option allows you to specify constraints on the time gaps that separate item sets. If selected, item sets with time gaps smaller than the Minimum gap or larger than the Maximum gap that you specify will not be considered to form part of a sequence. Use this option to avoid counting sequences that include long time intervals or those that take place in a very short time span.

Note: If the time field used is a time, date, or timestamp field, the time gap is expressed in seconds. For numeric fields, the time gap is expressed in the same units as the time field.

For example, consider the following list of transactions.

Table 1. Example list of transactions

ID	Time	Content
1001	1	apples
1001	2	bread
1001	5	cheese
1001	6	dressing

If you build a model on these data with the minimum gap set to 2, you would get the following sequences:

`apples -> cheese`

`apples -> dressing`

`bread -> cheese`

`bread -> dressing`

You would not see sequences such as `apples -> bread` because the gap between `apples` and `bread` is smaller than the minimum gap. Similarly, consider the following alternative data.

Table 2. Example list of transactions

ID	Time	Content
1001	1	apples
1001	2	bread
1001	5	cheese
1001	20	dressing

If the maximum gap were set to 10, you would not see any sequences with `dressing`, because the gap between `cheese` and `dressing` is too large for them to be considered part of the same sequence.

Related information

- [Sequence node](#)
- [Sequence Node Fields Options](#)
- [Sequence Node Model Options](#)

Sequence Model Nuggets

Sequence model nuggets represent the sequences found for a particular output field discovered by the Sequence node and can be added to streams to generate predictions.

When you run a stream containing a Sequence node, the node adds a pair of fields containing predictions and associated confidence values for each prediction from the sequence model to the data. By default, three pairs of fields containing the top three predictions (and their associated confidence values) are added. You can change the number of predictions generated when you build the model by setting the Sequence node model options at build time, as well as on the Settings tab after adding the model nugget to a stream. See the topic [Sequence Model Nugget Settings](#) for more information.

The new field names are derived from the model name. The field names are `$S-sequence-n` for the prediction field (where *n* indicates the *n*th prediction) and `$SC-sequence-n` for the confidence field. In a stream with multiple Sequence Rules nodes in a series, the new field names will include numbers in the prefix to distinguish them from each other. The first Sequence Set node in the stream will use the usual names, the second node will use names starting with `$S1-` and `$SC1-`, the third node will use names starting with `$S2-` and `$SC2-`, and so on. Predictions are displayed in order by confidence, so that `$S-sequence-1` contains the prediction with the highest confidence, `$S-sequence-2` contains the prediction with the next highest confidence, and so on. For records where the number of available predictions is smaller than the number of predictions requested, remaining predictions contain the value `$null$`. For example, if only two predictions can be made for a particular record, the values of `$S-sequence-3` and `$SC-sequence-3` will be `$null$`.

For each record, the rules in the model are compared to the set of transactions processed for the current ID so far, including the current record and any previous records with the same ID and earlier timestamp. The k rules with the highest confidence values that apply to this set of transactions are used to generate the k predictions for the record, where k is the number of predictions specified on the Settings tab after adding the model to the stream. (If multiple rules predict the same outcome for the transaction set, only the rule with the highest confidence is used.) See the topic [Sequence Model Nugget Settings](#) for more information.

As with other types of association rule models, the data format must match the format used in building the sequence model. For example, models built using tabular data can be used to score only tabular data. See the topic [Scoring Association Rules](#) for more information.

Note: When scoring data using a generated Sequence Set node in a stream, any tolerance or gap settings that you selected in building the model are ignored for scoring purposes.

Predictions from Sequence Rules

The node handles the records in a time-dependent manner (or order-dependent, if no timestamp field was used to build the model). Records should be sorted by the ID field and timestamp field (if present). However, predictions are not tied to the timestamp of the record to which they are added. They simply refer to the most likely items to occur *at some point in the future*, given the history of transactions for the current ID up to the current record.

Note that the predictions for each record do not necessarily depend on that record's transactions. If the current record's transactions do not trigger a specific rule, rules will be selected based on the previous transactions for the current ID. In other words, if the current record doesn't add any useful predictive information to the sequence, the prediction from the last useful transaction for this ID is carried forward to the current record.

For example, suppose you have a Sequence model with the single rule

Jam → Bread (0.66)

and you pass it the following records.

Table 1. Example records

ID	Purchase	Prediction
001	jam	bread
001	milk	bread

Notice that the first record generates a prediction of *bread*, as you would expect. The second record also contains a prediction of *bread*, because there's no rule for *jam* followed by *milk*; therefore, the *milk* transaction doesn't add any useful information, and the rule **Jam → Bread** still applies.

Generating New Nodes

The Generate menu allows you to create new SuperNodes based on the sequence model.

- **Rule SuperNode.** Creates a SuperNode that can detect and count occurrences of sequences in scored data. This option is disabled if no rule is selected. See the topic [Generating a Rule SuperNode from a Sequence Model Nugget](#) for more information.
- **Model to Palette.** Returns the model to the Models palette. This is useful in situations where a colleague may have sent you a stream containing the model and not the model itself.
- [Sequence Model Nugget Details](#)
- [Sequence Model Nugget Settings](#)
- [Sequence Model Nugget Summary](#)
- [Generating a Rule SuperNode from a Sequence Model Nugget](#)

Related information

- [Generating a Rule Set from an Association Model Nugget](#)
 - [Generating a Filtered Model](#)
 - [Sequence Model Nugget Details](#)
 - [Sequence Model Nugget Settings](#)
 - [Sequence Model Nugget Summary](#)
 - [Generating a Rule SuperNode from a Sequence Model Nugget](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
 - [Sequence node](#)
-

Sequence Model Nugget Details

The Model tab for a Sequence model nugget displays the rules extracted by the algorithm. Each row in the table represents a rule, with the antecedent (the "if" part of the rule) in the first column followed by the consequent (the "then" part of the rule) in the second column.

Each rule is shown in the following format.

Table 1. Rule format

Antecedent	Consequent
beer and cannedveg	beer
fish fish	fish

The first example rule is interpreted as *for IDs that had "beer" and "cannedveg" in the same transaction, there is likely a subsequent occurrence of "beer."* The second example rule can be interpreted as *for IDs that had "fish" in one transaction and then "fish" in another, there is a likely subsequent occurrence of "fish."* Note that in the first rule, *beer* and *cannedveg* are purchased at the same time; in the second rule, *fish* is purchased in two separate transactions.

Sort menu. The Sort menu button on the toolbar controls the sorting of rules. Direction of sorting (ascending or descending) can be changed using the sort direction button (up or down arrow).

You can sort rules by:

- Support %
- Confidence %
- Rule Support %
- Consequent
- First Antecedent
- Last Antecedent
- Number of Items (antecedents)

For example, the following table is sorted in descending order by number of items. Rules with multiple items in the antecedent set precede those with fewer items.

Table 2. Rules sorted by number of items

Antecedent	Consequent
beer and cannedveg and frozenmeal	frozenmeal
beer and cannedveg	beer
fish fish	fish
softdrink	softdrink

Show/hide criteria menu. The Show/hide criteria menu button (grid icon) controls options for the display of rules. The following display options are available:

- **Instances** displays information about the number of unique IDs for which the *full sequence*—both antecedents and consequent—occurs. (Note this differs from Association models, for which the number of instances refers to the number of IDs for which *only* the antecedents apply.) For example, given the rule **bread** → **cheese**, the number of IDs in the training data that include both *bread* and *cheese* are referred to as **instances**.
- **Support** displays the proportion of IDs in the training data for which the antecedents are true. For example, if 50% of the training data includes the antecedent *bread* then the support for the **bread** → **cheese** rule would be 50%. (Unlike Association models, support is not based on the number of instances, as noted earlier.)
- **Confidence** displays the percentage of the IDs where a correct prediction is made, out of all the IDs for which the rule makes a prediction. It is calculated as the number of IDs for which the entire sequence is found divided by the number of IDs for which the antecedents are found, based on the training data. For example, if 50% of the training data contains *cannedveg* (indicating antecedent support) but only 20% contains both *cannedveg* and *frozenmeal*, then confidence for the rule **cannedveg** → **frozenmeal** would be **Rule Support / Antecedent Support** or, in this case, 40%.
- **Rule Support** for Sequence models is based on instances and displays the proportion of training records for which the entire rule, antecedents, and consequent(s), are true. For example, if 20% of the training data contains both *bread* and *cheese*, then rule support for the rule **bread** → **cheese** is 20%.

Note that the proportions are based on valid transactions (transactions with at least one observed item or true value) rather than total transactions. Invalid transactions—those with no items or true values—are discarded for these calculations.

Filter button. The Filter button (funnel icon) on the menu expands the bottom of the dialog box to show a panel where active rule filters are displayed. Filters are used to narrow the number of rules displayed on the Models tab.

Figure 1. Filter button



To create a filter, click the Filter icon to the right of the expanded panel. This opens a separate dialog box in which you can specify constraints for displaying rules. Note that the Filter button is often used in conjunction with the Generate menu to first filter rules and then generate a model containing that subset of rules. For more information, see [Specifying Filters for Rules](#) below.

Related information

- [Generating a Rule Set from an Association Model Nugget](#)
 - [Generating a Filtered Model](#)
 - [Sequence Model Nuggets](#)
 - [Sequence Model Nugget Settings](#)
 - [Sequence Model Nugget Summary](#)
 - [Generating a Rule SuperNode from a Sequence Model Nugget](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Sequence Model Nugget Settings

The Settings tab for a Sequence model nugget displays scoring options for the model. This tab is available only after the model has been added to the stream canvas for scoring.

Maximum number of predictions. Specify the maximum number of predictions included for each set of basket items. The rules with the highest confidence values that apply to this set of transactions are used to generate predictions for the record up to the specified limit.

Related information

- [Generating a Rule Set from an Association Model Nugget](#)
 - [Generating a Filtered Model](#)
 - [Sequence Model Nuggets](#)
 - [Sequence Model Nugget Details](#)
 - [Sequence Model Nugget Summary](#)
 - [Generating a Rule SuperNode from a Sequence Model Nugget](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Sequence Model Nugget Summary

The Summary tab for a sequence rule model nugget displays the number of rules discovered and the minimum and maximum for support and confidence in the rules. If you have executed an Analysis node attached to this modeling node, information from that analysis will also be displayed in this section.

See the topic [Browsing model nuggets](#) for more information.

Related information

- [Generating a Rule Set from an Association Model Nugget](#)
 - [Generating a Filtered Model](#)
 - [Sequence Model Nuggets](#)
 - [Sequence Model Nugget Details](#)
 - [Sequence Model Nugget Settings](#)
 - [Generating a Rule SuperNode from a Sequence Model Nugget](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Generating a Rule SuperNode from a Sequence Model Nugget

To generate a rule SuperNode based on a sequence rule:

1. On the Model tab for the sequence rule model nugget, click on a row in the table to select the desired rule.
2. From the rule browser menus choose:
Generate > Rule SuperNode

Important: To use the generated SuperNode, you must sort the data by ID field (and Time field, if any) before passing them into the SuperNode. The SuperNode will not detect sequences properly in unsorted data.

You can specify the following options for generating a rule SuperNode:

Detect. Specifies how matches are defined for data passed into the SuperNode.

- **Antecedents only.** The SuperNode will identify a match any time it finds the antecedents for the selected rule in the correct order within a set of records having the same ID, regardless of whether the consequent is also found. Note that this does not take into account timestamp tolerance or item gap constraint settings from the original Sequence modeling node. When the last antecedent item set is detected in the stream (and all other antecedents have been found in the proper order), all subsequent records with the current ID will contain the summary selected below.
- **Entire sequence.** The SuperNode will identify a match any time it finds the antecedents and the consequent for the selected rule in the correct order within a set of records having the same ID. This does not take into account timestamp tolerance or item gap constraint settings from the original Sequence modeling node. When the consequent is detected in the stream (and all antecedents have also been found in the correct order), the current record and all subsequent records with the current ID will contain the summary selected below.

Display. Controls how match summaries are added to the data in the Rule SuperNode output.

- **Consequent value for first occurrence.** The value added to the data is the consequent value predicted based on the first occurrence of the match. Values are added as a new field named *rule_n_consequent*, where *n* is the rule number (based on the order of creation of Rule SuperNodes in the stream).
- **True value for first occurrence.** The value added to the data is true if there is at least one match for the ID and false if there is no match. Values are added as a new field named *rule_n_flag*.
- **Count of occurrences.** The value added to the data is the number of matches for the ID. Values are added as a new field named *rule_n_count*.
- **Rule number.** The value added is the rule number for the selected rule. **Rule numbers** are assigned based on the order in which the SuperNode was added to the stream. For example, the first Rule SuperNode is considered *rule 1*, the second Rule SuperNode is considered *rule 2*, etc. This option is most useful when you will be including multiple Rule SuperNodes in your stream. Values are added as a new field named *rule_n_number*.
- **Include confidence figures.** If selected, this option will add the rule confidence to the data stream as well as the selected summary. Values are added as a new field named *rule_n_confidence*.

Related information

- [Generating a Rule Set from an Association Model Nugget](#)
 - [Generating a Filtered Model](#)
 - [Sequence Model Nuggets](#)
 - [Sequence Model Nugget Details](#)
 - [Sequence Model Nugget Settings](#)
 - [Sequence Model Nugget Summary](#)
-

Association Rules node

Association rules are statements of the following form.

```
if condition(s) then prediction(s)
```

For example, "If a customer purchases a razor and after shave, then that customer will purchase shaving cream with 80% confidence." The Association Rules node extracts a set of rules from the data, pulling out the rules with the highest information content. The Association Rules node is very similar to the Apriori node, however, there are some notable differences:

- The Association Rules node cannot process transactional data.
- The Association Rules node can process data that has the List storage type and the Collection measurement level.
- The Association Rules node can be used with IBM® SPSS® Analytic Server. This provides scalability and means that you can process big data and take advantage of faster parallel processing.
- The Association Rules node provides additional settings, such as the ability to restrict the number of rules that are generated, thereby increasing the processing speed.
- Output from the model nugget is shown in the Output Viewer.

Note: The Association Rules node does not support the Model Evaluation or Champion Challenger steps in IBM SPSS Collaboration and Deployment Services.

Note: The Association Rules node ignores empty records when building a model if the field type is a flag. Empty records are records where all the fields used in the model build have a false value.

A stream that shows a working example of using Association Rules, named geospatial_association.str, and which references the data files InsuranceData.sav, CountyData.sav, and ChicagoAreaCounties.ships available from the Demos directory of your IBM SPSS Modeler installation. You can access the Demos directory from the IBM SPSS Modeler program group on the Windows Start menu. The geospatial_association.str file is in the streams directory.

- [Association Rules - Fields Options](#)
 - [Association Rules - Rule building](#)
 - [Association Rules - Transformations](#)
 - [Association Rules - Output](#)
 - [Association Rules - Model Options](#)
 - [Association Rules Model Nuggets](#)
-

Association Rules - Fields Options

On the Fields tab, you choose whether you want to use the field role settings that are already defined in upstream nodes, such as a previous Type node, or make the field assignments manually.

Use predefined roles

This option uses the role settings (such as targets, or predictors) from an upstream Type node (or the Types tab of an upstream source node). Fields with an input role are considered to be Conditions, fields with a target role are considered to be Predictions, and those fields that are used as inputs and targets are considered to have both roles.

Use custom field assignments

Choose this option if you want to assign targets, predictors, and other roles manually on this screen.

Fields

If you selected Use custom field assignments, use the arrow buttons to assign items manually from this list to the boxes on the right of the screen. The icons indicate the valid measurement levels for each field.

Both (Condition or Prediction)

Fields added to this list can take either the condition or prediction role in rules that are generated by the model. This is on a rule by rule basis, so a field might be a condition in one rule and a prediction in another.

Prediction only

Fields added to this list can appear only as a prediction (also known as a "consequent") of a rule. The presence of a field in this list does not mean that the field is used in any rules, only that if it is used it can be only a prediction.

Condition only

Fields added to this list can appear only as a condition (also known as an "antecedent") of a rule. The presence of a field in this list does not mean that the field is used in any rules, only that if it is used it can be only a condition.

Related information

- [Association Rules - Rule building](#)
 - [Association Rules - Transformations](#)
 - [Association Rules - Output](#)
 - [Association Rules - Model Options](#)
-

Association Rules - Rule building

Items per rule

Use these options to specify how many items, or values, can be used in each rule.

Note: The combined total of these two fields cannot exceed 10.

Maximum conditions

Select the maximum number of conditions that can be included in a single rule.

Maximum predictions

Select the maximum number of predictions that can be included in a single rule.

Rule building

Use these options to specify the number and type of rules to build.

Maximum number of rules

Specify the maximum number of rules that can be considered for use in the building of rules for your model.
Rule criterion for top N

Select the criterion that is used to establish which are the top N rules, where N is the value that is entered in the Maximum number of rules field. You can choose from the following criterion.

- Confidence
- Rule Support
- Condition Support
- Lift
- Deployability

Only true values for flags

When your data is in tabular format, select this option to include only true values for flag fields in the resulting rules. Selecting true values can help make rules easier to understand. The option does not apply to data in transactional format. For more information, see [Tabular versus Transactional Data](#).

Rule criterion

If you select Enable rule criterion, you can use these options to select the minimum strength that rules must meet to be considered for use in your model.

- Confidence Specify the minimum percentage value for the Confidence level for a rule that is produced by the model. If the model produces a rule with a level less than this amount, the rule is discarded.
- Rule Support Specify the minimum percentage value for the Rule Support level for a rule that is produced by the model. If the model produces a rule with a level less than this amount, the rule is discarded.
- Condition Support Specify the minimum percentage value for the Condition Support level for a rule that is produced by the model. If the model produces a rule with a level less than the specified amount, the rule is discarded.
- Lift Specify the minimum Lift value that is allowed for a rule that is produced by the model. If the model produces a rule with a value less than the specified amount, the rule is discarded.

Exclude rules

In some cases, the association between two or more fields is known or is self-evident, in such a case you can exclude rules where the fields predict each other. By excluding rules that contain both values, you reduce irrelevant input and increase the chances of finding useful results.

Fields

Select the associated fields that you do not want to use together in rule building. For example, associated fields might be Car Manufacturer and Car Model, or School Year and Age of Pupil. When the model creates rules, if the rule contains at least one of the fields selected on either side of the rule (condition or prediction), the rule is discarded.

Related information

- [Association Rules - Fields Options](#)
- [Association Rules - Transformations](#)
- [Association Rules - Output](#)
- [Association Rules - Model Options](#)

Association Rules - Transformations

Binning

Use these options to specify how continuous (numeric range) fields are binned.

Number of bins

Any continuous fields set to be automatically binned are divided into the number of equally spaced bins that you specify. You can select any number in the range 2 - 10.

List fields

Maximum list length

To restrict the number of items to be included in the model if the length of a list field is unknown, enter the maximum length of the list. You can select any number in the range 1 - 100. If a list is longer than the number you enter, the model will still use the field but only include values up to this number; any extra values in the field are ignored.

Related information

- [Association Rules - Fields Options](#)
 - [Association Rules - Rule building](#)
 - [Association Rules - Output](#)
 - [Association Rules - Model Options](#)
-

Association Rules - Output

Use the options in this pane to control what output is generated when the model is built.

Rules tables

Use these options to create one or more table types that display the best number of rules (based on a number you specify) for each selected criterion.

Confidence

Confidence is the ratio of rule support to condition support. Of the items with the listed condition values, the percentage that has the predicted consequent values. Creates a table that contains the best N association rules that are based on confidence to be included in the output (where N is the Rules to display value).

Rule Support

The proportion of items for which the entire rule, conditions, and predictions are true. For all items in the dataset, the percentage that is correctly accounted for and predicted by the rule. This measure gives an overall importance of the rule. Creates a table that contains the best N association rules that are based on rule support to be included in the output (where N is the Rules to display value).

Lift

The ratio of rule confidence and the prior probability of having the prediction. The ratio of the Confidence value for a rule versus the percentage the Consequent values occur in the overall population. This ratio gives a measure of how well the rule improves over chance. Creates a table that contains the best N association rules that are based on lift to be included in the output (where N is the Rules to display value).

Condition Support

The proportion of items for which the conditions are true. Creates a table that contains the best N association rules that are based on antecedent support to be included in the output (where N is the Rules to display value).

Deployability

A measure of what percentage of the training data satisfies the condition but not the prediction. This measure shows how often the rule misses. It is effectively the opposite of Confidence. Creates a table that contains the best N association rules that are based on deployability to be included in the output (N is the Rules to display value).

Rules to display

Set the maximum number of rules to display in the tables.

Model information tables

Use one or more of these options to select which model tables to include in the output.

- Field Transformations
- Records Summary
- Rule Statistics
- Most Frequent Values
- Most Frequent Fields

Sortable word cloud of rules.

Use these options to create a word cloud that displays the rules outputs. Words are displayed in increasing text sizes to indicate their importance.

Create a sortable word cloud.

Check this box to create a sortable word cloud in your output.

Default sort

Select the sort type to be used to when initially creating the word cloud. The word cloud is interactive and you can change the criterion in the Model Viewer to see different rules and sorts. You can choose from the following sort options:

- Confidence.
- Rule Support
- Lift
- Condition Support.

- Deployability

Max rules to display

Set the number of rules to be displayed in the word cloud; the maximum you can choose is 20.

Related information

- [Association Rules - Fields Options](#)
 - [Association Rules - Rule building](#)
 - [Association Rules - Transformations](#)
 - [Association Rules - Model Options](#)
-

Association Rules - Model Options

Use the settings on this tab to specify the scoring options for Association Rules models.

Model name You can generate the model name that is automatically based on the target field (or model type in cases where no such field is specified), or specify a custom name.

Maximum number of predictions Specify the maximum number of predictions that are included in the score result. This option is used with the Rule Criterion entries to produce the “top” predictions, where “top” indicates the highest level of confidence, support, lift, and so on.

Rule Criterion Select the measure that is used to determine the strength of the rules. Rules are sorted by the strength of criteria that are selected here in order to return the top predictions for an item set. You can choose from 5 different criteria.

- Confidence Confidence is the ratio of rule support to condition support. Of the items with the listed condition values, the percentage that has the predicted consequent values.
- Condition Support The proportion of items for which the conditions are true.
- Rule Support The proportion of items for which the entire rule, conditions, and predictions are true. Calculated by multiplying the Condition Support value by the Confidence value.
- Lift The ratio of rule confidence and the prior probability of having the prediction.
- Deployability A measure of what percentage of the training data satisfies the condition but not the prediction.

Allow repeat predictions To include multiple rules with the same prediction during scoring, select this check box. For example, selecting this enables the following rules to be scored.

```
bread & cheese -> wine
cheese & fruit -> wine
```

Note: Rules with multiple predictions (`bread & cheese & fruit -> wine & pate`) are considered repeat predictions only if all predictions (`wine & pate`) were predicted before.

Only score rules when predictions are not present in the input To ensure that predictions are not also present in the input, select this option. For example, if the purpose of scoring is to make a home furniture product recommendation, then it is unlikely that input that already contains a dining room table is likely to purchase another. In such a case, select this option. However, if products are perishable or disposable (such as cheese, baby formula, or tissue), then rules where the consequent is already present in the input might be of value. In the latter case, the most useful option might be Score all rules.

Only score rules when predictions are present in the input To ensure that predictions are also present in the input, select this option. This approach is useful when you are attempting to gain insight into existing customers or transactions. For example, you might want to identify rules with the highest lift and then explore which customers fit these rules.

Score all rules To include all rules during scoring, regardless of the presence or absence of predictions, select this option.

Related information

- [Association Rules - Fields Options](#)
 - [Association Rules - Rule building](#)
 - [Association Rules - Transformations](#)
 - [Association Rules - Output](#)
-

Association Rules Model Nuggets

The model nugget contains information about the rules extracted from your data during model building.

Viewing Results

You can browse the rules generated by Association Rules models by using the Model tab on the dialog box. Browsing a model nugget shows you the information about the rules before generating new nodes or scoring the model.

Scoring the Model

Refined model nuggets may be added to a stream and used for scoring. See the topic [Using Model Nuggets in Streams](#) for more information. Model nuggets used for scoring include an extra Settings tab on their respective dialog boxes. See the topic [Association Rules Model Nugget Settings](#) for more information.

- [Association Rules Model Nugget Details](#)
- [Association Rules Model Nugget Settings](#)

Related information

- [Association Rule model nugget details](#)
- [Specifying Filters for Rules](#)
- [Generating Graphs for Rules](#)
- [Association Rule Model Nugget Summary](#)
- [Generating a Rule Set from an Association Model Nugget](#)
- [Generating a Filtered Model](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)

Association Rules Model Nugget Details

The Association Rules model nugget displays details of the model in the Model tab of the Output Viewer. For more information on using the viewer, see [Working with output](#).

The GSAR modeling operation creates a number of new fields with the prefix \$A as shown in the following table.

Table 1. New fields created by the Association Rules modeling operation

Field name	Description
\$A-<prediction>#	This field contains the prediction from the model for the scored records. <prediction> is the name of the field included in the Predictions role in the model, and # is a sequence of numbers for the output rules (for example, if the score is set to include 3 rules the sequence of numbers will be from 1 to 3).
\$AC-<prediction>#	This field contains the confidence in the prediction. <prediction> is the name of the field included in the Predictions role in the model, and # is a sequence of numbers for the output rules (for example, if the score is set to include 3 rules the sequence of numbers will be from 1 to 3).
\$A-Rule_ID#	This column contains the ID of the rule predicted for each record in the scored data set. # is a sequence of numbers for the output rules (for example, if the score is set to include 3 rules the sequence of numbers will be from 1 to 3).

Related information

- [Association Rules Model Nuggets](#)
- [Association Rules Model Nugget Settings](#)
- [Model Nuggets](#)
- [The models palette](#)
- [Using Model Nuggets in Streams](#)
- [Browsing model nuggets](#)

Association Rules Model Nugget Settings

The Settings tab for an Association Rules model nugget displays scoring options for the model. This tab is available only after the model is added to the stream canvas for scoring.

Maximum number of predictions Specify the maximum number of predictions that are included for each set of items. The rules with the highest confidence values that apply to this set of transactions are used to generate predictions for the record up to the specified limit. Use this option with the Rule Criterion option to produce the “top” predictions, where *top* indicates the highest level of confidence, support, lift, and so on.

Rule Criterion Select the measure that is used to determine the strength of rules. Rules are sorted by the strength of criteria that are selected here in order to return the top predictions for an item set. You can choose from the following criteria.

- Confidence
- Rule Support
- Lift
- Condition Support
- Deployability

Allow repeat predictions To include multiple rules with the same consequent when scoring, select this check box. For example, selecting this option means that the following rule can be scored:

```
bread & cheese -> wine  
cheese & fruit -> wine
```

To exclude repeat predictions when scoring, clear the check box.

Note: Rules with multiple consequents (`bread & cheese & fruit -> wine & pate`) are considered repeat predictions only if all consequents (`wine & pate`) have been predicted before.

Only score rules when predictions are not present in the input Select to ensure that consequents are not also present in the input. For example, if the purpose of scoring is to make a home furniture product recommendation, then it is unlikely that input that already contains a dining room table is likely to purchase another one. In such a case, select this option. On the other hand, if products are perishable or disposable (such as cheese, baby formula, or tissue), then rules where the consequent is already present in the input might be of value. In the latter case, the most useful option might be Score all rules.

Only score rules when predictions are present in the input Select this option to ensure that consequents are also present in the input. This approach is useful when you are attempting to gain insight into existing customers or transactions. For example, you might want to identify rules with the highest lift and then explore which customers fit these rules.

Score all rules To include all rules when scoring, regardless of the presence or absence of consequents in the input, select this option.

Related information

- [Association Rules Model Nuggets](#)
 - [Association Rules Model Nugget Details](#)
 - [Model Nuggets](#)
 - [The models palette](#)
 - [Using Model Nuggets in Streams](#)
 - [Browsing model nuggets](#)
-

Time Series Models

- [Why forecast?](#)
 - [Time series data](#)
 - [Predictor series](#)
 - [Spatio-Temporal Prediction modeling node](#)
 - [TCM Node](#)
 - [Time Series node](#)
 - [Time Series Node \(deprecated\)](#)
-

Why forecast?

To forecast means to predict the values of one or more series over time. For example, you may want to predict the expected demand for a line of products or services in order to allocate resources for manufacturing or distribution. Because planning decisions take time to implement, forecasts are an essential tool in many planning processes.

Methods of modeling time series assume that history repeats itself—if not exactly, then closely enough that by studying the past, you can make better decisions in the future. To predict sales for next year, for example, you would probably start by looking at this year's sales and work backward to figure out what trends or patterns, if any, have developed in recent years. But patterns can be difficult to gauge. If your sales increase several weeks in a row, for example, is this part of a seasonal cycle or the beginning of a long-term trend?

Using statistical modeling techniques, you can analyze the patterns in your past data and project those patterns to determine a range within which future values of the series are likely to fall. The result is more accurate forecasts on which to base your decisions.

Time series data

A **time series** is an ordered collection of measurements taken at regular intervals—for example, daily stock prices or weekly sales data. The measurements may be of anything that interests you, and each series can generally be classified as one of the following:

- Dependent. A series that you want to forecast.
- Predictor. A series that may help to explain the target—for example, using an advertising budget to predict sales. Predictors can only be used with ARIMA models.
- Event. A special predictor series used to account for predictable recurring incidents—for example, sales promotions.
- Intervention. A special predictor series used to account for one-time past incidents—for example, a power outage or employee strike.

The intervals can represent any unit of time, but the interval must be the same for all measurements. Moreover, any interval for which there is no measurement must be set to the missing value. Thus, the number of intervals with measurements (including those with missing values) defines the length of time of the historical span of the data.

- [Characteristics of time series](#)
- [Autocorrelation and partial autocorrelation functions](#)
- [Series transformations](#)

Characteristics of time series

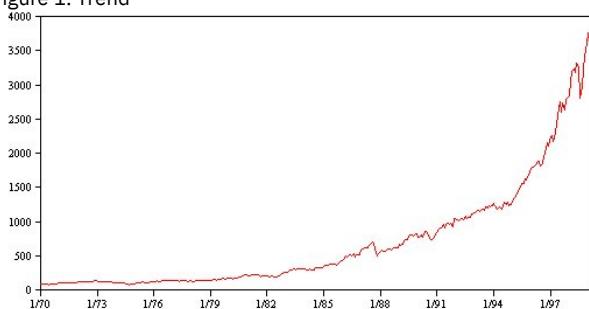
Studying the past behavior of a series will help you identify patterns and make better forecasts. When plotted, many time series exhibit one or more of the following features:

- Trends
 - Seasonal and nonseasonal cycles
 - Pulses and steps
 - Outliers
- [Trends](#)
 - [Seasonal cycles](#)
 - [Nonseasonal cycles](#)
 - [Pulses and steps](#)
 - [Outliers](#)

Trends

A **trend** is a gradual upward or downward shift in the level of the series or the tendency of the series values to increase or decrease over time.

Figure 1. Trend



Trends are either **local** or **global**, but a single series can exhibit both types. Historically, series plots of the stock market index show an upward global trend. Local downward trends have appeared in times of recession, and local upward trends have appeared in times of prosperity.

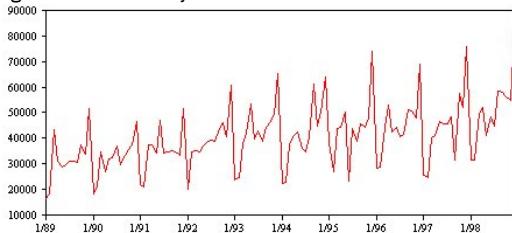
Trends can also be either **linear** or **nonlinear**. Linear trends are positive or negative additive increments to the level of the series, comparable to the effect of simple interest on principal. Nonlinear trends are often multiplicative, with increments that are proportional to the previous series value(s).

Global linear trends are fit and forecast well by both exponential smoothing and ARIMA models. In building ARIMA models, series showing trends are generally differenced to remove the effect of the trend.

Seasonal cycles

A **seasonal cycle** is a repetitive, predictable pattern in the series values.

Figure 1. Seasonal cycle



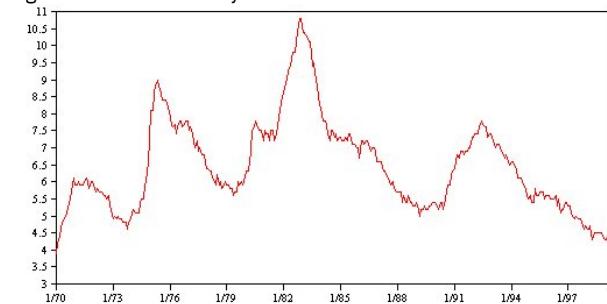
Seasonal cycles are tied to the interval of your series. For instance, monthly data typically cycles over quarters and years. A monthly series might show a significant quarterly cycle with a low in the first quarter or a yearly cycle with a peak every December. Series that show a seasonal cycle are said to exhibit **seasonality**.

Seasonal patterns are useful in obtaining good fits and forecasts, and there are exponential smoothing and ARIMA models that capture seasonality.

Nonseasonal cycles

A **nonseasonal cycle** is a repetitive, possibly unpredictable, pattern in the series values.

Figure 1. Nonseasonal cycle



Some series, such as unemployment rate, clearly display cyclical behavior; however, the periodicity of the cycle varies over time, making it difficult to predict when a high or low will occur. Other series may have predictable cycles but do not neatly fit into the Gregorian calendar or have cycles longer than a year. For example, the tides follow the lunar calendar, international travel and trade related to the Olympics swell every four years, and there are many religious holidays whose Gregorian dates change from year to year.

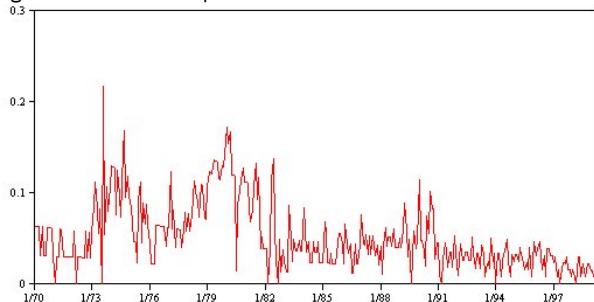
Nonseasonal cyclical patterns are difficult to model and generally increase uncertainty in forecasting. The stock market, for example, provides numerous instances of series that have defied the efforts of forecasters. All the same, nonseasonal patterns must be accounted for when they exist. In many cases, you can still identify a model that fits the historical data reasonably well, which gives you the best chance to minimize uncertainty in forecasting.

Pulses and steps

Many series experience abrupt changes in level. They generally come in two types:

- A sudden, *temporary* shift, or **pulse**, in the series level
- A sudden, *permanent* shift, or **step**, in the series level

Figure 1. Series with a pulse



When steps or pulses are observed, it is important to find a plausible explanation. Time series models are designed to account for gradual, not sudden, change. As a result, they tend to underestimate pulses and be ruined by steps, which lead to poor model fits and uncertain forecasts. (Some instances of seasonality may appear to exhibit sudden changes in level, but the level is constant from one seasonal period to the next.)

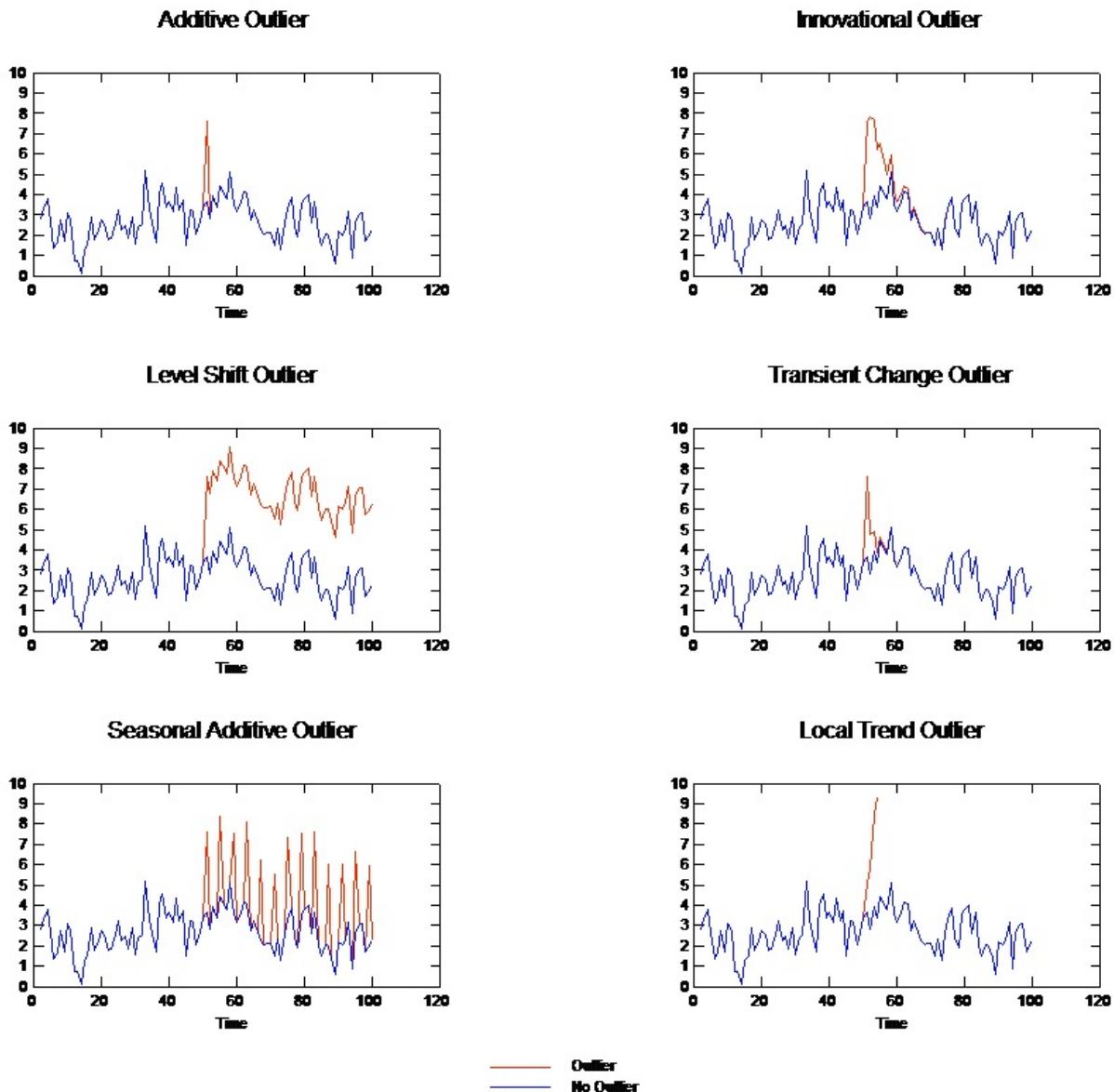
If a disturbance can be explained, it can be modeled using an **intervention** or **event**. For example, during August 1973, an oil embargo imposed by the Organization of Petroleum Exporting Countries (OPEC) caused a drastic change in the inflation rate, which then returned to normal levels in the ensuing months. By specifying a **point intervention** for the month of the embargo, you can improve the fit of your model, thus indirectly improving your forecasts. For example, a retail store might find that sales were much higher than usual on the day all items were marked 50% off. By specifying the 50%-off promotion as a recurring **event**, you can improve the fit of your model and estimate the effect of repeating the promotion on future dates.

Outliers

Shifts in the level of a time series that cannot be explained are referred to as **outliers**. These observations are inconsistent with the remainder of the series and can dramatically influence the analysis and, consequently, affect the forecasting ability of the time series model.

The following figure displays several types of outliers commonly occurring in time series. The blue lines represent a series without outliers. The red lines suggest a pattern that might be present if the series contained outliers. These outliers are all classified as **deterministic** because they affect only the mean level of the series.

Figure 1. Outlier types



- **Additive Outlier.** An additive outlier appears as a surprisingly large or small value occurring for a single observation. Subsequent observations are unaffected by an additive outlier. Consecutive additive outliers are typically referred to as **additive outlier patches**.
- **Innovational Outlier.** An innovational outlier is characterized by an initial impact with effects lingering over subsequent observations. The influence of the outliers may increase as time proceeds.
- **Level Shift Outlier.** For a level shift, all observations appearing after the outlier move to a new level. In contrast to additive outliers, a level shift outlier affects many observations and has a permanent effect.
- **Transient Change Outlier.** Transient change outliers are similar to level shift outliers, but the effect of the outlier diminishes exponentially over the subsequent observations. Eventually, the series returns to its normal level.
- **Seasonal Additive Outlier.** A seasonal additive outlier appears as a surprisingly large or small value occurring repeatedly at regular intervals.
- **Local Trend Outlier.** A local trend outlier yields a general drift in the series caused by a pattern in the outliers after the onset of the initial outlier.

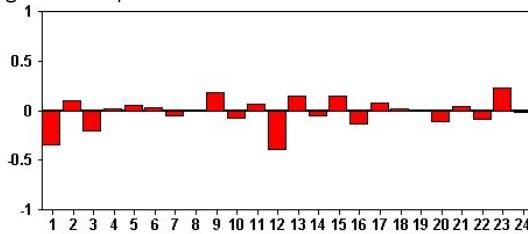
Outlier detection in time series involves determining the location, type, and magnitude of any outliers present. Tsay (1988) proposed an iterative procedure for detecting mean level change to identify deterministic outliers. This process involves comparing a time series model that assumes no outliers are present to another model that incorporates outliers. Differences between the models yield estimates of the effect of treating any given point as an outlier.

Autocorrelation and partial autocorrelation functions

Autocorrelation and partial autocorrelation are measures of association between current and past series values and indicate which past series values are most useful in predicting future values. With this knowledge, you can determine the order of processes in an ARIMA model. More specifically,

- Autocorrelation function (ACF). At lag k , this is the correlation between series values that are k intervals apart.
- Partial autocorrelation function (PACF). At lag k , this is the correlation between series values that are k intervals apart, accounting for the values of the intervals between.

Figure 1. ACF plot for a series



The x axis of the ACF plot indicates the lag at which the autocorrelation is computed; the y axis indicates the value of the correlation (between -1 and 1). For example, a spike at lag 1 in an ACF plot indicates a strong correlation between each series value and the preceding value, a spike at lag 2 indicates a strong correlation between each value and the value occurring two points previously, and so on.

- A positive correlation indicates that large current values correspond with large values at the specified lag; a negative correlation indicates that large current values correspond with small values at the specified lag.
- The absolute value of a correlation is a measure of the strength of the association, with larger absolute values indicating stronger relationships.

Series transformations

Transformations are often useful for stabilizing a series before estimating models. This is particularly important for ARIMA models, which require series to be **stationary** before models are estimated. A series is stationary if the global level (mean) and average deviation from the level (variance) are constant throughout the series.

While most interesting series are not stationary, ARIMA is effective as long as the series can be made stationary by applying transformations, such as the natural log, differencing, or seasonal differencing.

Variance stabilizing transformations. Series in which the variance changes over time can often be stabilized using a natural log or square root transformation. These are also called functional transformations.

- Natural log. The natural logarithm is applied to the series values.
- Square root. The square root function is applied to the series values.

Natural log and square root transformations cannot be used for series with negative values.

Level stabilizing transformations. A slow decline of the values in the ACF indicates that each series value is strongly correlated with the previous value. By analyzing the change in the series values, you obtain a stable level.

- Simple differencing. The differences between each value and the previous value in the series are computed, except the oldest value in the series. This means that the differenced series will have one less value than the original series.
- Seasonal differencing. Identical to simple differencing, except that the differences between each value and the previous seasonal value are computed.

When either simple or seasonal differencing is simultaneously in use with either the log or square root transformation, the variance stabilizing transformation is always applied first. When simple and seasonal differencing are both in use, the resulting series values are the same whether simple differencing or seasonal differencing is applied first.

Predictor series

Predictor series include related data that may help explain the behavior of the series to be forecast. For example, a Web- or catalog-based retailer might forecast sales based on the number of catalogs mailed, the number of phone lines open, or the number of hits to the company Web page.

Any series can be used as a predictor provided that the series extends as far into the future as you want to forecast and has complete data with no missing values.

Use care when adding predictors to a model. Adding large numbers of predictors will increase the time required to estimate models. While adding predictors may improve a model's ability to fit the historical data, it doesn't necessarily mean that the model does a better job of forecasting, so the added complexity may not be worth the trouble. Ideally, the goal should be to identify the simplest model that does a good job of forecasting.

As a general rule, it is recommended that the number of predictors should be less than the sample size divided by 15 (at most, one predictor per 15 cases).

Predictors with missing data. Predictors with incomplete or missing data cannot be used in forecasting. This applies to both historical data and future values. In some cases, you can avoid this limitation by setting the model's estimation span to exclude the oldest data when estimating models.

Spatio-Temporal Prediction modeling node

Spatio-Temporal prediction (STP) has many potential applications such as energy management for buildings or facilities, performance analysis and forecasting for mechanical service engineers, or public transport planning. In these applications measurements such as energy usage are often taken over space and time. Questions that might be relevant to the recording of these measurements include what factors will affect future observations, what can be done to effect a wanted change, or to better manage the system? To address these questions, you can use statistical techniques that can forecast future values at different locations, and can explicitly model adjustable factors to perform what-if analysis.

STP analysis uses data that contains location data, input fields for prediction (predictors), a time field, and a target field. Each location has numerous rows in the data that represent the values of each predictor at each time of measurement. After the data is analyzed, it can be used to predict target values at any location within the shape data that is used in the analysis. It can also forecast when input data for the future points in time are known.

Note: The STP node does not support the Model Evaluation or Champion Challenger steps in IBM® SPSS® Collaboration and Deployment Services. A stream that shows a worked example of using STP, named stp_server_demo.str, and which references the data files room_data.csv and score_data.csv is available from the Demos directory of your IBM SPSS Modeler installation. You can access the Demos directory from the IBM SPSS Modeler program group on the Windows Start menu. The stp_server_demo.str file is in the streams directory.

- [Spatio-Temporal Prediction - Fields Options](#)
 - [Spatio-Temporal Prediction - Time Intervals](#)
 - [Spatio-Temporal Prediction - Basic Build Options](#)
 - [Spatio-Temporal Prediction - Advanced Build Options](#)
 - [Spatio-Temporal Prediction - Output](#)
 - [Spatio-Temporal Prediction - Model Options](#)
 - [Spatio-Temporal Prediction Model Nugget](#)
-

Spatio-Temporal Prediction - Fields Options

On the Fields tab, you choose whether you want to use the field role settings that are already defined in upstream nodes, or make the field assignments manually.

Use predefined roles

This option uses the role settings (targets and predictors only) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments

To assign targets, predictors, and other roles manually on this screen, select this option.

Fields

Displays all of the fields in the data that can be selected. Use the arrow buttons to assign items manually from this list to the various boxes on the right of the screen. The icons indicate the valid measurement levels for each field.

Note: STP requires 1 record per location, per time interval to function correctly; therefore these are mandatory fields.

At the bottom of the Fields pane, either click the All button to select all fields, regardless of measurement level, or click an individual measurement level button to select all fields with that measurement level.

Target

Select one field as the target for the prediction.

Note: You can only select fields with a measurement level of continuous.

Location

Select the location type to be used in the model.

Note: You can only select fields with a measurement level of geospatial.

Location label

Shape data often includes a field that shows the names of the features in the layer, for example, this might be the names of states or counties. Use this field to associate a name, or label, with a location by selecting a categorical field to label the chosen Location field in your output.

Time Field

Select the time fields to use in your predictions.

Note: You can select only fields with a measurement level of continuous and a storage type of time, date, timestamp, or integer.

Predictors (Inputs)

Choose one or more fields as inputs for the prediction.

Note: You can select only fields with a measurement level of continuous.

Related information

- [Spatio-Temporal Prediction - Time Intervals](#)
 - [Spatio-Temporal Prediction - Basic Build Options](#)
 - [Spatio-Temporal Prediction - Advanced Build Options](#)
 - [Spatio-Temporal Prediction - Output](#)
 - [Spatio-Temporal Prediction - Model Options](#)
-

Spatio-Temporal Prediction - Time Intervals

In the Time Intervals pane, you can select the options to set the time interval and any required aggregation over time.

Data preparation is required to convert time fields into an index before you can build an STP model; for conversion to be possible the time field must have a constant interval between records. If your data does not already contain this information, use the options in this pane to set this interval before you can use the modeling node.

Time interval Select the interval that you want the data set to be converted to. Available options depend on the storage type of the field that is chosen as the Time Field for the model on the Fields tab.

- Periods Only available for integer time fields; this is a series of intervals, with a uniform interval between each measurement, that does not match any of the other available intervals.
- Years Only available for Date or Timestamp time fields.
- Quarters Only available for Date or Timestamp time fields. If you choose this option, you are prompted to select the Start month of the first quarter.
- Months Only available for Date or Timestamp time fields.
- Weeks Only available for Date or Timestamp time fields.
- Days Only available for Date or Timestamp time fields.
- Hours Only available for Time or Timestamp time fields.
- Minutes Only available for Time or Timestamp time fields.
- Seconds Only available for Time or Timestamp time fields.

When you select the Time interval, you are prompted to complete further fields. The fields available depend on both the time interval and the storage type. The fields that might be displayed are shown in the following list.

- Number of days per week
- Number of hours in a day
- Week begins on The first day of the week
- Day begins at The time at which you consider a new day to start.
- Interval value You can choose one of the following options: 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, or 30.
- Start month The month on which the financial year starts.
- Starting period If you are using Periods, select the starting period.

Data matches specified time interval settings If your data already contains the correct time intervals information and does not need to be converted, select this check box. When you select this box, the fields in the Aggregation area are unavailable.

Aggregation

Only available if you clear the Data matches specified time interval settings check box; specify the options for aggregating fields to match the specified interval. For example, if you have a mix of weekly and monthly data, you might aggregate, or "roll up", the weekly values to achieve a uniform monthly interval. Select the default settings to be used for aggregation of different field types and create any custom settings that you want for any specific fields.

- **Continuous** Set the default aggregation method to be applied to all continuous fields that are not individually specified. You can choose from several methods:
 - Sum
 - Mean
 - Minimum
 - Maximum
 - Median

- First quartile
- Third quartile

Custom settings for specified fields To apply a specific aggregation function to individual fields, select them in this table and choose the aggregation method.

- Field Use the Add field button to display the Select Fields dialog box and choose the required fields. The chosen fields are displayed in this column.
- Aggregation function From the drop-down list, select the aggregation function to convert the field to the specified time interval.

Related information

- [Spatio-Temporal Prediction - Fields Options](#)
- [Spatio-Temporal Prediction - Basic Build Options](#)
- [Spatio-Temporal Prediction - Advanced Build Options](#)
- [Spatio-Temporal Prediction - Output](#)
- [Spatio-Temporal Prediction - Model Options](#)

Spatio-Temporal Prediction - Basic Build Options

Use the settings in this dialog box to set the basic model building options.

Model Settings

Include intercept

Including the intercept (the constant term in the model) can increase the overall accuracy of the solution. If you can assume the data passes through the origin, you can exclude the intercept.

Maximum autoregressive order

Autoregressive orders specify which previous values are used to predict current values. Use this option to specify the number of previous records that are used to calculate a new value. You can choose any integer between 1 and 5.

Spatial Covariance

Estimation method

Select the estimation method to be used; you can choose either Parametric or Nonparametric. For the Parametric method you can choose from one of three Model types:

- Gaussian
- Exponential
- Powered Exponential If you select this option, you must also specify the Power level to be used. This level can be any value between 1 and 2, changed in 0.1 increments.

Related information

- [Spatio-Temporal Prediction - Fields Options](#)
- [Spatio-Temporal Prediction - Time Intervals](#)
- [Spatio-Temporal Prediction - Advanced Build Options](#)
- [Spatio-Temporal Prediction - Output](#)
- [Spatio-Temporal Prediction - Model Options](#)

Spatio-Temporal Prediction - Advanced Build Options

Users with detailed knowledge of STP can use the following options to fine-tune the model building process.

Maximum percentage of missing values

Specify the maximum percentage of records that contain missing values that can be included in the model.

Significance level for hypotheses testing in model build

Specify the significance level value to be used for all tests of STP model estimation, including two Goodness of Fit tests, Effect F-tests and Coefficient T-tests. This level can be any value from 0 to 1, changed in 0.01 increments.

Related information

- [Spatio-Temporal Prediction - Fields Options](#)
 - [Spatio-Temporal Prediction - Time Intervals](#)
 - [Spatio-Temporal Prediction - Basic Build Options](#)
 - [Spatio-Temporal Prediction - Advanced Build Options](#)
 - [Spatio-Temporal Prediction - Model Options](#)
-

Spatio-Temporal Prediction - Output

Before you build the model, use the options on this pane to select the output that you want to include in the output viewer.

Model Information

Model specifications

Select this option to include model specification information in the model output.

Temporal information summary

Select this option to include a summary of temporal information in the model output.

Evaluation

Model quality

Select this option to include model quality in the model output.

Test of effects in mean structure model

Select this option to include test of effects information in the model output.

Interpretation

Mean structure model coefficients

Select this option to include mean structure model coefficients information in the model output.

Autoregressive coefficients

Select this option to include autoregressive coefficients information in the model output.

Tests of decay over space

Select this option to include test of spatial covariance, or decay over space, information in the model output.

Parametric spatial covariance model parameters plot

Select this option to include parametric spatial covariance model parameter plot information in the model output.

Note: This option is only available if you selected the Parametric estimation method on the Basics tab.

Correlations heat map

Select this option to include a map of the target values in the model output.

Note: If there are more than 500 locations in your model, the map output will not be created.

Correlations map

Select this option to include a map of correlations in the model output.

Note: If there are more than 500 locations in your model, the map output will not be created.

Location clusters

Select this option to include location clustering output in the model output. Only output which does not require access to map data is included as part of the cluster output.

Note: This output can only be created for a non-parametric spatial covariance model.

If you choose this option, you can set the following:

- Similarity threshold Select the threshold at which output clusters are to be considered as similar enough to be merged into a single cluster.
- Maximum number of clusters to display Set the upper limit for the number of clusters that can be included in the model output.

Related information

- [Spatio-Temporal Prediction - Fields Options](#)
- [Spatio-Temporal Prediction - Time Intervals](#)
- [Spatio-Temporal Prediction - Basic Build Options](#)
- [Spatio-Temporal Prediction - Advanced Build Options](#)
- [Spatio-Temporal Prediction - Model Options](#)

Spatio-Temporal Prediction - Model Options

Model Name You can generate the model name automatically, based on the target fields, or specify a custom name. The automatically generated name is the target field name.

Uncertainty factor (%) Uncertainty factor is a percentage value that represents the growth in uncertainty when you forecast into the future. The upper and lower limit of forecast uncertainty increase by this percentage each step into the future. Set the uncertainty factor to be applied to your model outputs; this sets the upper and lower boundaries for the predicted values.

Related information

- [Spatio-Temporal Prediction - Fields Options](#)
- [Spatio-Temporal Prediction - Time Intervals](#)
- [Spatio-Temporal Prediction - Basic Build Options](#)
- [Spatio-Temporal Prediction - Advanced Build Options](#)
- [Spatio-Temporal Prediction - Output](#)

Spatio-Temporal Prediction Model Nugget

The Spatio-Temporal Prediction (STP) model nugget displays details of the model in the Model tab of the Output Viewer. For more information on using the viewer, see [Working with output](#).

The spatio-temporal prediction (STP) modeling operation creates a number of new fields with the prefix \$STP- as shown in the following table.

Table 1. New fields created by the STP modeling operation

Field name	Description
\$STP-<Time>	The time field created as part of the model build. The settings on the Time Intervals pane of the Build Options tab determine how this field is created. <Time> is the original name of the field selected as the Time Field on the Fields tab. Note: This field is only created if you converted the original Time Field as part of the model build.
\$STP-<Target>	This field contains the predictions for the target value. <Target> is the name of the original Target field for the model
\$STPVAR-<Target>	This field contains the VarianceOfPointPrediction values. <Target> is the name of the original Target field for the model
\$STPLCI-<Target>	This field contains the LowerOfPredictionInterval values; that is, the lower bound of confidence. <Target> is the name of the original Target field for the model
\$STPUCI-<Target>	This field contains the UpperOfPredictionInterval values; that is, the upper bound of confidence. <Target> is the name of the original Target field for the model

- [Spatio-Temporal Prediction Model Settings](#)

Spatio-Temporal Prediction Model Settings

Use the Settings tab to control the level of uncertainty that you deem acceptable in the modeling operation.

Uncertainty factor (%) Uncertainty factor is a percentage value that represents the growth in uncertainty when you forecast into the future. The upper and lower limit of forecast uncertainty increase by this percentage each step into the future. Set the uncertainty factor to be applied to your model outputs; this sets the upper and lower boundaries for the predicted values.

TCM Node

Use this node to create a temporal causal model (TCM).

- [Temporal Causal Models](#)
 - [TCM Model Nugget](#)
 - [Temporal Causal Model Scenarios](#)
-

Temporal Causal Models

Temporal causal modeling attempts to discover key causal relationships in time series data. In temporal causal modeling, you specify a set of target series and a set of candidate inputs to those targets. The procedure then builds an autoregressive time series model for each target and includes only those inputs that have a causal relationship with the target. This approach differs from traditional time series modeling where you must explicitly specify the predictors for a target series. Since temporal causal modeling typically involves building models for multiple related time series, the result is referred to as a *model system*.

In the context of temporal causal modeling, the term *causal* refers to Granger causality. A time series X is said to "Granger cause" another time series Y if regressing for Y in terms of past values of both X and Y results in a better model for Y than regressing only on past values of Y.

Note: The Temporal causal modeling node does not support the Model Evaluation or Champion Challenger steps in IBM® SPSS® Collaboration and Deployment Services.

Examples

Business decision makers can use temporal causal modeling to uncover causal relationships within a large set of time-based metrics that describe the business. The analysis might reveal a few controllable inputs, which have the largest impact on key performance indicators.

Managers of large IT systems can use temporal causal modeling to detect anomalies in a large set of interrelated operational metrics. The causal model then allows going beyond anomaly detection and discovering the most likely root causes of the anomalies.

Field requirements

There must be at least one target. By default, fields with a predefined role of None are not used.

Data structure

Temporal causal modeling supports two types of data structures.

Column-based data

For column-based data, each time series field contains the data for a single time series. This structure is the traditional structure of time series data, as used by the Time Series Modeler.

Multidimensional data

For multidimensional data, each time series field contains the data for multiple time series. Separate time series, within a particular field, are then identified by a set of values of categorical fields referred to as *dimension* fields. For example, sales data for two different sales channels (retail and web) might be stored in a single *sales* field. A dimension field that is named *channel*, with values 'retail' and 'web', identifies the records that are associated with each of the two sales channels.

Note: To build a temporal causal model, you need enough data points. The product uses the constraint:

$m > (L + KL + 1)$

where m is the number of data points, L is the number of lags, and K is the number of predictors. Make sure your data set is big enough so that the number of data points (m) satisfies the condition.

- [Time Series to Model \(Temporal Causal Modeling\)](#)
 - [Observations \(Temporal Causal Modeling\)](#)
 - [Time Interval for Analysis \(Temporal Causal Modeling\)](#)
 - [Aggregation and Distribution \(Temporal Causal Modeling\)](#)
 - [Missing Values \(Temporal Causal Modeling\)](#)
 - [General Data Options \(Temporal Causal Modeling\)](#)
 - [General Build Options \(Temporal Causal Modeling\)](#)
 - [Series to Display \(Temporal Causal Modeling\)](#)
 - [Output Options \(Temporal Causal Modeling\)](#)
 - [Estimation Period \(Temporal Causal Modeling\)](#)
 - [Model Options \(Temporal Causal Modeling\)](#)
 - [Interactive Output \(Temporal Causal Modeling\)](#)
-

Time Series to Model (Temporal Causal Modeling)

On the Fields tab, use the Time Series settings to specify the series to include in the model system.

Select the option for the data structure that applies to your data. For multidimensional data, click Select Dimensions to specify the dimension fields. The specified order of the dimension fields defines the order in which they appear in all subsequent dialogs and output. Use the up and down arrow buttons on the Select Dimensions subdialog to reorder the dimension fields.

For column-based data, the term *series* has the same meaning as the term *field*. For multidimensional data, fields that contain time series are referred to as *metric* fields. A time series, for multidimensional data, is defined by a metric field and a value for each of the dimension fields. The following considerations apply to both column-based and multidimensional data.

- Series that are specified as candidate inputs or as both target and input are considered for inclusion in the model for each target. The model for each target always includes lagged values of the target itself.
- Series that are specified as forced inputs are always included in the model for each target.
- At least one series must be specified as either a target or as both target and input.
- When Use predefined roles is selected, fields that have a role of Input are set as candidate inputs. No predefined role maps to a forced input. For more information, see the topic [Roles](#).

Multidimensional data

For multidimensional data, you specify metric fields and associated roles in a grid, where each row in the grid specifies a single metric and role. By default, the model system includes series for all combinations of the dimension fields for each row in the grid. For example, if there are dimensions for *region* and *brand* then, by default, specifying the metric *sales* as a target means that there is a separate sales target series for each combination of *region* and *brand*.

For each row in the grid, you can customize the set of values for any of the dimension fields by clicking the ellipsis button for a dimension. This action opens the Select Dimension Values subdialog. You can also add, delete, or copy grid rows.

The Series Count column displays the number of sets of dimension values that are currently specified for the associated metric. The displayed value can be larger than the actual number of series (one series per set). This condition occurs when some of the specified combinations of dimension values do not correspond to series contained by the associated metric.

- [Select Dimension Values \(Temporal Causal Modeling\)](#)

Select Dimension Values (Temporal Causal Modeling)

For multidimensional data, you can customize the analysis by specifying which dimension values apply to a particular metric field with a particular role. For example, if *sales* is a metric field and *channel* is a dimension with values 'retail' and 'web', you can specify that 'web' sales is an input and 'retail' sales is a target. You can also specify dimension subsets that apply to all metric fields used in the analysis. For example, if *region* is a dimension field that indicates geographical region, then you can limit the analysis to particular regions.

All values

Specifies that all values of the current dimension field are included. This option is the default.

Select values to include or exclude

Use this option to specify the set of values for the current dimension field. When Include is selected for the Mode, only values that are specified in the Selected values list are included. When Exclude is selected for the Mode, all values other than the values that are specified in the Selected values list are included.

You can filter the set of values from which to choose. Values that meet the filter condition appear in the Matched tab and values that do not meet the filter condition appear in the Unmatched tab of the Unselected values list. The All tab lists all unselected values, regardless of any filter condition.

- You can use asterisks (*) to indicate wildcard characters when you specify a filter.
- To clear the current filter, specify an empty value for the search term on the Filter Displayed Values dialog.

Observations (Temporal Causal Modeling)

On the Fields tab, use the Observations settings to specify the fields that define the observations.

Observations that are defined by date/times

You can specify that the observations are defined by a date, time, or timestamp field. In addition to the field that defines the observations, select the appropriate time interval that describes the observations. Depending on the specified time interval, you can also specify other settings, such as the interval between observations (increment) or the number of days per week. The following considerations apply to the time interval:

- Use the value Irregular when the observations are irregularly spaced in time, such as the time at which a sales order is processed. When Irregular is selected, you must specify the time interval that is used for the analysis, from the Time Interval settings on the

Data Specifications tab.

- When the observations represent a date and time and the time interval is hours, minutes, or seconds, then use Hours per day, Minutes per day, or Seconds per day. When the observations represent a time (duration) without reference to a date and the time interval is hours, minutes, or seconds, then use Hours (non-periodic), Minutes (non-periodic), or Seconds (non-periodic).
- Based on the selected time interval, the procedure can detect missing observations. Detecting missing observations is necessary since the procedure assumes that all observations are equally spaced in time and that there are no missing observations. For example, if the time interval is Days and the date 2014-10-27 is followed by 2014-10-29, then there is a missing observation for 2014-10-28. Values are imputed for any missing observations. Settings for handling missing values can be specified from the Data Specifications tab.
- The specified time interval allows the procedure to detect multiple observations in the same time interval that need to be aggregated together and to align observations on an interval boundary, such as the first of the month, to ensure that the observations are equally spaced. For example, if the time interval is Months, then multiple dates in the same month are aggregated together. This type of aggregation is referred to as *grouping*. By default, observations are summed when grouped. You can specify a different method for grouping, such as the mean of the observations, from the Aggregation and Distribution settings on the Data Specifications tab.
- For some time intervals, the additional settings can define breaks in the normal equally spaced intervals. For example, if the time interval is Days, but only weekdays are valid, you can specify that there are five days in a week, and the week begins on Monday.

Observations that are defined by periods or cyclic periods

Observations can be defined by one or more integer fields that represent periods or repeating cycles of periods, up to an arbitrary number of cycle levels. With this structure, you can describe series of observations that don't fit one of the standard time intervals. For example, a fiscal year with only 10 months can be described with a cycle field that represents years and a period field that represents months, where the length of one cycle is 10.

Fields that specify cyclic periods define a hierarchy of periodic levels, where the lowest level is defined by the Period field. The next highest level is specified by a cycle field whose level is 1, followed by a cycle field whose level is 2, and so on. Field values for each level, except the highest, must be periodic with respect to the next highest level. Values for the highest level cannot be periodic. For example, in the case of the 10-month fiscal year, months are periodic within years and years are not periodic.

- The length of a cycle at a particular level is the periodicity of the next lowest level. For the fiscal year example, there is only one cycle level and the cycle length is 10 since the next lowest level represents months and there are 10 months in the specified fiscal year.
- Specify the starting value for any periodic field that does not start from 1. This setting is necessary for detecting missing values. For example, if a periodic field starts from 2 but the starting value is specified as 1, then the procedure assumes that there is a missing value for the first period in each cycle of that field.

Related information

- [Time Interval for Analysis \(Temporal Causal Modeling\)](#)
 - [Aggregation and Distribution \(Temporal Causal Modeling\)](#)
 - [Missing Values \(Temporal Causal Modeling\)](#)
 - [Estimation Period \(Temporal Causal Modeling\)](#)
-

Time Interval for Analysis (Temporal Causal Modeling)

The time interval that is used for the analysis can differ from the time interval of the observations. For example, if the time interval of the observations is Days, you might choose Months for the time interval for analysis. The data are then aggregated from daily to monthly data before the model is built. You can also choose to distribute the data from a longer to a shorter time interval. For example, if the observations are quarterly then you can distribute the data from quarterly to monthly data.

The available choices for the time interval at which the analysis is done depend on how the observations are defined and the time interval of those observations. In particular, when the observations are defined by cyclic periods, then only aggregation is supported. In that case, the time interval of the analysis must be greater than or equal to the time interval of the observations.

The time interval for the analysis is specified from the Time Interval settings on the Data Specifications tab. The method by which the data are aggregated or distributed is specified from the Aggregation and Distribution settings on the Data Specifications tab.

Related information

- [Observations \(Temporal Causal Modeling\)](#)
 - [Aggregation and Distribution \(Temporal Causal Modeling\)](#)
 - [Estimation Period \(Temporal Causal Modeling\)](#)
-

Aggregation and Distribution (Temporal Causal Modeling)

Aggregation functions

When the time interval that is used for the analysis is longer than the time interval of the observations, the input data are aggregated. For example, aggregation is done when the time interval of the observations is Days and the time interval for analysis is Months. The following aggregation functions are available: mean, sum, mode, min, or max.

Distribution functions

When the time interval that is used for the analysis is shorter than the time interval of the observations, the input data are distributed. For example, distribution is done when the time interval of the observations is Quarters and the time interval for analysis is Months. The following distribution functions are available: mean or sum.

Grouping functions

Grouping is applied when observations are defined by date/times and multiple observations occur in the same time interval. For example, if the time interval of the observations is Months, then multiple dates in the same month are grouped and associated with the month in which they occur. The following grouping functions are available: mean, sum, mode, min, or max. Grouping is always done when the observations are defined by date/times and the time interval of the observations is specified as Irregular.

Note: Although grouping is a form of aggregation, it is done before any handling of missing values whereas formal aggregation is done after any missing values are handled. When the time interval of the observations is specified as Irregular, aggregation is done only with the grouping function.

Aggregate cross-day observations to previous day

Specifies whether observations with times that cross a day boundary are aggregated to the values for the previous day. For example, for hourly observations with an eight-hour day that starts at 20:00, this setting specifies whether observations between 00:00 and 04:00 are included in the aggregated results for the previous day. This setting applies only if the time interval of the observations is Hours per day, Minutes per day or Seconds per day and the time interval for analysis is Days.

Custom settings for specified fields

You can specify aggregation, distribution, and grouping functions on a field by field basis. These settings override the default settings for the aggregation, distribution, and grouping functions.

Related information

- [Observations \(Temporal Causal Modeling\)](#)
 - [Time Interval for Analysis \(Temporal Causal Modeling\)](#)
 - [Missing Values \(Temporal Causal Modeling\)](#)
-

Missing Values (Temporal Causal Modeling)

Missing values in the input data are replaced with an imputed value. The following replacement methods are available:

Linear interpolation

Replaces missing values by using a linear interpolation. The last valid value before the missing value and the first valid value after the missing value are used for the interpolation. If the first or last observation in the series has a missing value, then the two nearest non-missing values at the beginning or end of the series are used.

Series mean

Replaces missing values with the mean for the entire series.

Mean of nearby points

Replaces missing values with the mean of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the mean.

Median of nearby points

Replaces missing values with the median of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the median.

Linear trend

This option uses all non-missing observations in the series to fit a simple linear regression model, which is then used to impute the missing values.

Other settings:

Maximum percentage of missing values (%)

Specifies the maximum percentage of missing values that are allowed for any series. Series with more missing values than the specified maximum are excluded from the analysis.

General Data Options (Temporal Causal Modeling)

Maximum number of distinct values per dimension field

This setting applies to multidimensional data and specifies the maximum number of distinct values that are allowed for any one dimension field. By default, this limit is set to 10000 but it can be increased to an arbitrarily large number.

General Build Options (Temporal Causal Modeling)

Confidence interval width (%)

This setting controls the confidence intervals for both forecasts and model parameters. You can specify any positive value less than 100. By default, a 95% confidence interval is used.

Maximum number of inputs for each target

This setting specifies the maximum number of inputs that are allowed in the model for each target. You can specify an integer in the range 1 - 20. The model for each target always includes lagged values of itself, so setting this value to 1 specifies that the only input is the target itself.

Model tolerance

This setting controls the iterative process that is used for determining the best set of inputs for each target. You can specify any value that is greater than zero. The default is 0.001. Model tolerance is a stop criterion for predictor selection. It can affect the number of predictors that are included in the final model. But if a target can predict itself very well, other predictors may not be included in the final model.

Some trial and error may be required (for example, if you have this value set to high, you can try setting it to a smaller value to see if other predictors can be included or not).

Outlier threshold (%)

An observation is flagged as an outlier if the probability, as calculated from the model, that it is an outlier exceeds this threshold. You can specify a value in the range 50 - 100.

Number of Lags for Each Input

This setting specifies the number of lag terms for each input in the model for each target. By default, the number of lag terms is automatically determined from the time interval that is used for the analysis. For example, if the time interval is months (with an increment of one month) then the number of lags is 12. Optionally, you can explicitly specify the number of lags. The specified value must be an integer in the range 1 - 20.

Continue estimation using existing models

If you already generated a temporal causal model, select this option to reuse the criteria settings that are specified for that model, rather than building a new model. In this way, you can save time by reestimating and producing a new forecast that is based on the same model settings as before but using more recent data.

Series to Display (Temporal Causal Modeling)

These options specify the series (targets or inputs) for which output is displayed. The content of the output for the specified series is determined by the Output Options settings.

Display targets associated with best-fitting models

By default, output is displayed for the targets that are associated with the 10 best-fitting models, as determined by the R square value. You can specify a different fixed number of best-fitting models or you can specify a percentage of best-fitting models. You can also choose from the following goodness of fit measures:

R square

Goodness-of-fit measure of a linear model, sometimes called the coefficient of determination. It is the proportion of variation in the target variable explained by the model. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well.

Root mean square percentage error

A measure of how much the model-predicted values differ from the observed values of the series. It is independent of the units that are used and can therefore be used to compare series with different units.

Root mean square error

The square root of mean square error. A measure of how much a dependent series varies from its model-predicted level, expressed in the same units as the dependent series.

BIC

Bayesian Information Criterion. A measure for selecting and comparing models based on the -2 reduced log likelihood. Smaller values indicate better models. The BIC also "penalizes" overparameterized models (complex models with a large number of inputs, for example), but more strictly than the AIC.

AIC

Akaike Information Criterion. A measure for selecting and comparing models based on the -2 reduced log likelihood. Smaller values indicate better models. The AIC "penalizes" overparameterized models (complex models with a large number of inputs, for example).

Specify Individual Series

You can specify individual series for which you want output.

- For column-based data, you specify the fields that contain the series that you want. The order of the specified fields defines the order in which they appear in the output.
- For multidimensional data, you specify a particular series by adding an entry to the grid for the metric field that contains the series. You then specify the values of the dimension fields that define the series.

- You can enter the value for each dimension field directly into the grid or you can select from the list of available dimension values. To select from the list of available dimension values, click the ellipsis button in the cell for the dimension that you want. This action opens the Select Dimension Value subdialog.
- You can search the list of dimension values, on the Select Dimension Value subdialog, by clicking the binoculars icon and specifying a search term. Spaces are treated as part of the search term. Asterisks (*) in the search term do not indicate wildcard characters.
- The order of the series in the grid defines the order in which they appear in the output.

For both column-based data and multidimensional data, output is limited to 30 series. This limit includes individual series (inputs or targets) that you specify and targets that are associated with best-fitting models. Individually specified series take precedence over targets that are associated with best-fitting models.

Related information

- [Output Options \(Temporal Causal Modeling\)](#)
-

Output Options (Temporal Causal Modeling)

These options specify the content of the output. Options in the Output for targets group generate output for the targets that are associated with the best-fitting models on the Series to Display settings. Options in the Output for series group generate output for the individual series that are specified on the Series to Display settings.

Overall model system

Displays a graphical representation of the causal relations between series in the model system. Tables of both model fit statistics and outliers for the displayed targets are included as part of the output item. When this option is selected in the Output for series group, a separate output item is created for each individual series that is specified on the Series to Display settings.

Causal relations between series have an associated significance level, where a smaller significance level indicates a more significant connection. You can choose to hide relations with a significance level that is greater than a specified value.

Model fit statistics and outliers

Tables of model fit statistics and outliers for the target series that are selected for display. These tables contain the same information as the tables in the Overall Model System visualization. These tables support all standard features for pivoting and editing tables.

Model effects and model parameters

Tables of model effects tests and model parameters for the target series that are selected for display. Model effects tests include the F statistic and associated significance value for each input included in the model.

Impact diagram

Displays a graphical representation of the causal relations between a series of interest and other series that it affects or that affect it. Series that affect the series of interest are referred to as *causes*. Selecting Effects generates an impact diagram that is initialized to display effects. Selecting Causes generates an impact diagram that is initialized to display causes. Selecting Both causes and effects generates two separate impact diagrams, one that is initialized to causes and one that is initialized to effects. You can interactively toggle between causes and effects in the output item that displays the impact diagram.

You can specify the number of levels of causes or effects to display, where the first level is just the series of interest. Each additional level shows more indirect causes or effects of the series of interest. For example, the third level in the display of effects consists of the series that contain series in the second level as a direct input. Series in the third level are then indirectly affected by the series of interest since the series of interest is a direct input to the series in the second level.

Series plot

Plots of observed and predicted values for the target series that are selected for display. When forecasts are requested, the plot also shows the forecasted values and the confidence intervals for the forecasts.

Residuals plot

Plots of the model residuals for the target series that are selected for display.

Top inputs

Plots of each displayed target, over time, along with the top 3 inputs for the target. The top inputs are the inputs with the lowest significance value. To accommodate different scales for the inputs and target, the y axis represents the z score for each series.

Forecast table

Tables of forecasted values and confidence intervals of those forecasts for the target series that are selected for display.

Outlier root cause analysis

Determines which series are most likely to be the cause of each outlier in a series of interest. Outlier root cause analysis is done for each target series that is included in the list of individual series on the Series to Display settings.

Output

Interactive outliers table and chart

Table and chart of outliers and root causes of those outliers for each series of interest. The table contains a single row for each outlier. The chart is an impact diagram. Selecting a row in the table highlights the path, in the impact diagram, from the

series of interest to the series that most likely causes the associated outlier.

Pivot table of outliers

Table of outliers and root causes of those outliers for each series of interest. This table contains the same information as the table in the interactive display. This table supports all standard features for pivoting and editing tables.

Causal levels

You can specify the number of levels to include in the search for the root causes. The concept of levels that is used here is the same as described for impact diagrams.

Model fit across all models

Histogram of model fit for all models and for selected fit statistics. The following fit statistics are available:

R square

Goodness-of-fit measure of a linear model, sometimes called the coefficient of determination. It is the proportion of variation in the target variable explained by the model. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well.

Root mean square percentage error

A measure of how much the model-predicted values differ from the observed values of the series. It is independent of the units that are used and can therefore be used to compare series with different units.

Root mean square error

The square root of mean square error. A measure of how much a dependent series varies from its model-predicted level, expressed in the same units as the dependent series.

BIC

Bayesian Information Criterion. A measure for selecting and comparing models based on the -2 reduced log likelihood. Smaller values indicate better models. The BIC also "penalizes" overparameterized models (complex models with a large number of inputs, for example), but more strictly than the AIC.

AIC

Akaike Information Criterion. A measure for selecting and comparing models based on the -2 reduced log likelihood. Smaller values indicate better models. The AIC "penalizes" overparameterized models (complex models with a large number of inputs, for example).

Outliers over time

Bar chart of the number of outliers, across all targets, for each time interval in the estimation period.

Series transformations

Table of any transformations that were applied to the series in the model system. The possible transformations are missing value imputation, aggregation, and distribution.

Related information

- [Interactive Output \(Temporal Causal Modeling\)](#)
 - [Series to Display \(Temporal Causal Modeling\)](#)
-

Estimation Period (Temporal Causal Modeling)

By default, the estimation period starts at the time of the earliest observation and ends at the time of the latest observation across all series.

By start and end times

You can specify both the start and end of the estimation period or you can specify just the start or just the end. If you omit the start or the end of the estimation period, the default value is used.

- If the observations are defined by a date/time field, then enter values for start and end in the same format that is used for the date/time field.
- For observations that are defined by cyclic periods, specify a value for each of the cyclic periods fields. Each field is displayed in a separate column.

By latest or earliest time intervals

Defines the estimation period as a specified number of time intervals that start at the earliest time interval or end at the latest time interval in the data, with an optional offset. In this context, the time interval refers to the time interval of the analysis. For example, assume that the observations are monthly but the time interval of the analysis is quarters. Specifying Latest and a value of 24 for the Number of time intervals means the latest 24 quarters.

Optionally, you can exclude a specified number of time intervals. For example, specifying the latest 24 time intervals and 1 for the number to exclude means that the estimation period consists of the 24 intervals that precede the last one.

Model Options (Temporal Causal Modeling)

Model Name

You can specify a custom name for the model or accept the automatically generated name, which is *TCM*.

Forecast

The option to Extend records into the future sets the number of time intervals to forecast beyond the end of the estimation period. The time interval in this case is the time interval of the analysis, which is specified on the Data Specifications tab. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period. There is no maximum limit for this setting.

Interactive Output (Temporal Causal Modeling)

The output from temporal causal modeling includes a number of interactive output objects. Interactive features are available by activating (double-clicking) the object in the Output Viewer.

Overall model system

Displays the causal relations between series in the model system. All lines that connect a particular target to its inputs have the same color. The thickness of the line indicates the significance of the causal connection, where thicker lines represent a more significant connection. Inputs that are not also targets are indicated with a black square.

- You can display relations for top models, a specified series, all series, or models with no inputs. The top models are the models that meet the criteria that were specified for best-fitting models on the Series to Display settings.
- You can generate impact diagrams for one or more series by selecting the series names in the chart, right-clicking, and then choosing Create Impact Diagram from the context menu.
- You can choose to hide causal relations that have a significance level that is greater than a specified value. Smaller significance levels indicate a more significant causal relation.
- You can display relations for a particular series by selecting the series name in the chart, right-clicking, and then choosing Highlight relations for series from the context menu.

Impact diagram

Displays a graphical representation of the causal relations between a series of interest and other series that it affects or that affect it. Series that affect the series of interest are referred to as *causes*.

- You can change the series of interest by specifying the name of the series that you want. Double-clicking any node in the impact diagram changes the series of interest to the series associated with that node.
- You can toggle the display between causes and effects and you can change the number of levels of causes or effects to display.
- Single-clicking any node opens a detailed sequence diagram for the series that is associated with the node.

Outlier root cause analysis

Determines which series are most likely to be the cause of each outlier in a series of interest.

- You can display the root cause for any outlier by selecting the row for the outlier in the Outliers table. You can also display the root cause by clicking the icon for the outlier in the sequence chart.
- Single-clicking any node opens a detailed sequence diagram for the series that is associated with the node.

Overall model quality

Histogram of model fit for all models, for a particular fit statistic. Clicking a bar in the bar chart filters the dot plot so that it displays only the models that are associated with the selected bar. You can find the model for a particular target series in the dot plot by specifying the series name.

Outlier distribution

Bar chart of the number of outliers, across all targets, for each time interval in the estimation period. Clicking a bar in the bar chart filters the dot plot so that it displays only the outliers that are associated with the selected bar.

Related information

- [Interactive output](#)
- [Output Options \(Temporal Causal Modeling\)](#)

TCM Model Nugget

The TCM modeling operation creates a number of new fields with the prefix \$TCM- as shown in the following table.

Table 1. New fields created by the TCM modeling operation

Field name	Description
\$TCM-colname	The value forecasted by the model for each target series.

\$TCMLCI-colname	The lower confidence intervals for each forecasted series.
\$TSUCI-colname	The upper confidence intervals for each forecasted series.
\$TCMResidual-colname	The noise residual value for each column of the generated model data.

- [TCM Model Nugget Settings](#)

TCM Model Nugget Settings

The Settings tab provides additional options for the TCM model nugget.

Forecast

The option to Extend records into the future sets the number of time intervals to forecast beyond the end of the estimation period. The time interval in this case is the time interval of the analysis, which is specified on the Data Specifications tab of the TCM node. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period.

Make available for scoring

Create new fields for each model to be scored. Enables you to specify the new fields to create for each model to be scored.

- Noises Residuals. If selected, creates a new field (with the default prefix \$TCM-) for the model residuals for each target field, together with a total of these values.
- Upper and lower confidence limits. If selected, creates new fields (with the default prefix \$TCM-) for the lower and upper confidence intervals, respectively, for each target field, together with totals of these values.

Targets included for scoring. Select available targets to include in the model score.

Temporal Causal Model Scenarios

The Temporal Causal Model Scenarios procedure runs user-defined scenarios for a temporal causal model system, with the data from the active dataset. A *scenario* is defined by a time series, that is referred to as the *root series*, and a set of user-defined values for that series over a specified time range. The specified values are then used to generate predictions for the time series that are affected by the root series. The procedure requires a model system file that was created by the Temporal Causal Modeling procedure. It is assumed that the active dataset is the same data as was used to create the model system file.

Example

Using the Temporal Causal Modeling procedure, a business decision maker discovered a key metric that affects a number of important performance indicators. The metric is controllable, so the decision maker wants to investigate the effect of various sets of values for the metric over the next quarter. The investigation is easily done by loading the model system file into the Temporal Causal Model Scenarios procedure and specifying the sets of values for the key metric.

- [Defining the Scenario Period \(Temporal Causal Model Scenarios\)](#)
- [Adding Scenarios and Scenario Groups \(Temporal Causal Model Scenarios\)](#)
- [Options \(Temporal Causal Model Scenarios\)](#)

Related information

- [Temporal Causal Models](#)

Defining the Scenario Period (Temporal Causal Model Scenarios)

The scenario period is the period over which you specify the values that are used to run your scenarios. It can start before or after the end of the estimation period. You can optionally specify to predict beyond the end of the scenario period. By default, predictions are generated through the end of the scenario period. All scenarios use the same scenario period and specifications for how far to predict.

Note: Predictions start at the first time period after the beginning of the scenario period. For example, if the scenario period starts on 2014-11-01 and the time interval is months, then the first prediction is for 2014-12-01.

Specify by start, end and predict through times

- If the observations are defined by a date/time field, then enter values for start, end, and predict through in the same format that is used for the date/time field. Values for date/time fields are aligned to the beginning of the associated time interval. For example, if the time interval of the analysis is months, then the value 10/10/2014 is adjusted to 10/01/2014, which is the beginning of the month.
- For observations that are defined by cyclic periods, specify a value for each of the cyclic periods fields. Each field is displayed in a separate column.

Specify by time intervals relative to end of estimation period

Defines start and end in terms of the number of time intervals relative to the end of the estimation period, where the time interval is the time interval of the analysis. The end of the estimation period is defined as time interval 0. Time intervals before the end of the estimation period have negative values and intervals after the end of the estimation period have positive values. You can also specify how many intervals to predict beyond the end of the scenario period. The default is 0.

For example, assume that the time interval of the analysis is months and that you specify 1 for starting interval, 3 for ending interval, and 1 for how far to predict beyond that. The scenario period is then the 3 months that follow the end of the estimation period. Predictions are generated for the second and third month of the scenario period and for 1 more month past the end of the scenario period.

Related information

- [Observations \(Temporal Causal Modeling\)](#)
 - [Time Interval for Analysis \(Temporal Causal Modeling\)](#)
-

Adding Scenarios and Scenario Groups (Temporal Causal Model Scenarios)

The Scenarios tab specifies the scenarios that are to be run. To define scenarios, you must first define the scenario period by clicking Define Scenario Period. Scenarios and scenario groups (applies only to multidimensional data) are created by clicking the associated Add Scenario or Add Scenario Group button. By selecting a particular scenario or scenario group in the associated grid, you can edit it, make a copy of it, or delete it.

Column-based data

The Root field column in the grid specifies the time series field whose values are replaced with the scenario values. The Scenario values column displays the specified scenario values in the order of earliest to latest. If the scenario values are defined by an expression, then the column displays the expression.

Multidimensional data

Individual Scenarios

Each row in the Individual Scenarios grid specifies a time series whose values are replaced by the specified scenario values. The series is defined by the combination of the field that is specified in the Root Metric column and the specified value for each of the dimension fields.

The content of the Scenario values column is the same as for column-based data.

Scenario Groups

A *scenario group* defines a set of scenarios that are based on a single root metric field and multiple sets of dimension values. Each set of dimension values (one value per dimension field), for the specified metric field, defines a time series. An individual scenario is then generated for each such time series, whose values are then replaced by the scenario values. Scenario values for a scenario group are specified by an expression, which is then applied to each time series in the group.

The Series Count column displays the number of sets of dimension values that are associated with a scenario group. The displayed value can be larger than the actual number of time series that are associated with the scenario group (one series per set). This condition occurs when some of the specified combinations of dimension values do not correspond to series contained by the root metric for the group.

As an example of a scenario group, consider a metric field *advertising* and two dimension fields *region* and *brand*. You can define a scenario group that is based on *advertising* as the root metric and that includes all combinations of *region* and *brand*. You might then specify *advertising*1.2* as the expression to investigate the effect of increasing *advertising* by 20 percent for each of the time series that are associated with the *advertising* field. If there are 4 values of *region* and 2 values of *brand*, then there are 8 such time series and thus 8 scenarios defined by the group.

- [Scenario Definition \(Temporal Causal Model Scenarios\)](#)
- [Scenario Group Definition \(Temporal Causal Model Scenarios\)](#)

Related information

- [Defining the Scenario Period \(Temporal Causal Model Scenarios\)](#)

Scenario Definition (Temporal Causal Model Scenarios)

Settings for defining a scenario depend on whether your data are column-based or multidimensional.

Root series

Specifies the root series for the scenario. Each scenario is based on a single root series. For column-based data, you select the field that defines the root series. For multidimensional data, you specify the root series by adding an entry to the grid for the metric field that contains the series. You then specify the values of the dimension fields that define the root series. The following apply to specifying the dimension values:

- You can enter the value for each dimension field directly into the grid or you can select from the list of available dimension values. To select from the list of available dimension values, click the ellipsis button in the cell for the dimension that you want. This action opens the Select Dimension Value subdialog.
- You can search the list of dimension values, on the Select Dimension Value subdialog, by clicking the binoculars icon and specifying a search term. Spaces are treated as part of the search term. Asterisks (*) in the search term do not indicate wildcard characters.

Specify affected targets

Use this option when you know specific targets that are affected by the root series and you want to investigate the effects on those targets only. By default, targets that are affected by the root series are automatically determined. You can specify the breadth of the series that are affected by the scenario with settings on the Options tab.

For column-based data, select the targets that you want. For multidimensional data, you specify target series by adding an entry to the grid for the target metric field that contains the series. By default, all series that are contained in the specified metric field are included. You can customize the set of included series by customizing the included values for one or more of the dimension fields. To customize the dimension values that are included, click the ellipsis button for the dimension that you want. This action opens the Select Dimension Values dialog.

The Series Count column (for multidimensional data) displays the number of sets of dimension values that are currently specified for the associated target metric. The displayed value can be larger than the actual number of affected target series (one series per set). This condition occurs when some of the specified combinations of dimension values do not correspond to series contained by the associated target metric.

Scenario ID

Each scenario must have a unique identifier. The identifier is displayed in output that is associated with the scenario. There are no restrictions, other than uniqueness, on the value of the identifier.

Specify scenario values for root series

Use this option to specify explicit values for the root series in the scenario period. You must specify a numeric value for each time interval that is listed in the grid. You can obtain the values of the root series (actual or forecasted) for each interval in the scenario period by clicking Read, Forecast, or Read\Forecast.

Specify expression for scenario values for root series

You can define an expression for computing the values of the root series in the scenario period. You can enter the expression directly or click the calculator button and create the expression from the Scenario Values Expression Builder.

- The expression can contain any target or input in the model system.
- When the scenario period extends beyond the existing data, the expression is applied to forecasted values of the fields in the expression.
- For multidimensional data, each field in the expression specifies a time series that is defined by the field and the dimension values that were specified for the root metric. It is those time series that are used to evaluate the expression.

As an example, assume that the root field is *advertising* and the expression is *advertising*1.2*. The values for *advertising* that are used in the scenario represent a 20 percent increase over the existing values.

Note: Scenarios are created by clicking Add Scenario on the Scenarios tab.

- [Select Dimension Values \(Temporal Causal Model Scenarios\)](#)

Select Dimension Values (Temporal Causal Model Scenarios)

For multidimensional data, you can customize the dimension values that define the targets that are affected by a scenario or scenario group. You can also customize the dimension values that define the set of root series for a scenario group.

All values

Specifies that all values of the current dimension field are included. This option is the default.

Select values

Use this option to specify the set of values for the current dimension field. You can filter the set of values from which to choose. Values that meet the filter condition appear in the Matched tab and values that do not meet the filter condition appear in the Unmatched tab of the Unselected values list. The All tab lists all unselected values, regardless of any filter condition.

- You can use asterisks (*) to indicate wildcard characters when you specify a filter.
- To clear the current filter, specify an empty value for the search term on the Filter Displayed Values dialog.

To customize dimension values for affected targets:

1. From the Scenario Definition or Scenario Group Definition dialog, select the target metric for which you want to customize dimension values.
2. Click the ellipsis button in the column for the dimension that you want to customize.

To customize dimension values for the root series of a scenario group:

1. From the Scenario Group Definition dialog, click the ellipsis button (in the root series grid) for the dimension that you want to customize.

Related information

- [Scenario Group Definition \(Temporal Causal Model Scenarios\)](#).

Scenario Group Definition (Temporal Causal Model Scenarios)

Root Series

Specifies the set of root series for the scenario group. An individual scenario is generated for each time series in the set. You specify the root series by adding an entry to the grid for the metric field that contains the series that you want. You then specify the values of the dimension fields that define the set. By default, all series that are contained in the specified root metric field are included. You can customize the set of included series by customizing the included values for one or more of the dimension fields. To customize the dimension values that are included, click the ellipsis button for a dimension. This action opens the Select Dimension Values dialog. The Series Count column displays the number of sets of dimension values that are currently included for the associated root metric. The displayed value can be larger than the actual number of root series for the scenario group (one series per set). This condition occurs when some of the specified combinations of dimension values do not correspond to series contained by the root metric.

Specify affected target series

Use this option when you know specific targets that are affected by the set of root series and you want to investigate the effects on those targets only. By default, targets that are affected by each root series are automatically determined. You can specify the breadth of the series that are affected by each individual scenario with settings on the Options tab.

You specify target series by adding an entry to the grid for the metric field that contains the series. By default, all series that are contained in the specified metric field are included. You can customize the set of included series by customizing the included values for one or more of the dimension fields. To customize the dimension values that are included, click the ellipsis button for the dimension that you want. This action opens the Select Dimension Values dialog.

The Series Count column displays the number of sets of dimension values that are currently specified for the associated target metric. The displayed value can be larger than the actual number of affected target series (one series per set). This condition occurs when some of the specified combinations of dimension values do not correspond to series contained by the associated target metric.

Scenario ID prefix

Each scenario group must have a unique prefix. The prefix is used to construct an identifier that is displayed in output that is associated with each individual scenario in the scenario group. The identifier for an individual scenario is the prefix, followed by an underscore, followed by the value of each dimension field that identifies the root series. The dimension values are separated by underscores. There are no restrictions, other than uniqueness, on the value of the prefix.

Expression for scenario values for root series

Scenario values for a scenario group are specified by an expression, which is then used to compute the values for each of the root series in the group. You can enter an expression directly or click the calculator button and create the expression from the Scenario Values Expression Builder.

- The expression can contain any target or input in the model system.
- When the scenario period extends beyond the existing data, the expression is applied to forecasted values of the fields in the expression.
- For each root series in the group, the fields in the expression specify time series that are defined by those fields and the dimension values that define the root series. It is those time series that are used to evaluate the expression. For example, if a root series is defined by `region='north'` and `brand='X'`, then the time series that are used in the expression are defined by those same dimension values.

As an example, assume that the root metric field is `advertising` and that there are two dimension fields `region` and `brand`. Also, assume that the scenario group includes all combinations of the dimension field values. You might then specify `advertising*1.2` as the expression to investigate the effect of increasing `advertising` by 20 percent for each of the time series that are associated with the `advertising` field.

Options (Temporal Causal Model Scenarios)

Maximum level for affected targets

Specifies the maximum number of levels of affected targets. Each successive level, up to the maximum of 5, includes targets that are more indirectly affected by the root series. Specifically, the first level includes targets that have the root series as a direct input. Targets in the second level have targets in the first level as a direct input, and so on. Increasing the value of this setting increases the complexity of the computation and might affect performance.

Maximum auto-detected targets

Specifies the maximum number of affected targets that are automatically detected for each root series. Increasing the value of this setting increases the complexity of the computation and might affect performance.

Impact diagram

Displays a graphical representation of the causal relations between the root series for each scenario and the target series that it affects.

Tables of both the scenario values and the predicted values for the affected targets are included as part of the output item. The graph includes plots of the predicted values of the affected targets. Single-clicking any node in the impact diagram opens a detailed sequence diagram for the series that is associated with the node. A separate impact diagram is generated for each scenario.

Series plots

Generates series plots of the predicted values for each of the affected targets in each scenario.

Forecast and scenario tables

Tables of predicted values and scenario values for each scenario. These tables contain the same information as the tables in the impact diagram. These tables support all standard features for pivoting and editing tables.

Include confidence intervals in plots and tables

Specifies whether confidence intervals for scenario predictions are included in both chart and table output.

Confidence interval width (%)

This setting controls the confidence intervals for the scenario predictions. You can specify any positive value less than 100. By default, a 95% confidence interval is used.

Time Series node

The Time Series node can be used with data in either a local or distributed environment; in a distributed environment you can harness the power of IBM® SPSS® Analytic Server. With this node, you can choose to estimate and build exponential smoothing, univariate Autoregressive Integrated Moving Average (ARIMA), or multivariate ARIMA (or transfer function) models for time series, and produce forecasts based on the time series data.

Exponential smoothing is a method of forecasting that uses weighted values of previous series observations to predict future values. As such, exponential smoothing is not based on a theoretical understanding of the data. It forecasts one point at a time, adjusting its forecasts as new data come in. The technique is useful for forecasting series that exhibit trend, seasonality, or both. You can choose from various exponential smoothing models that differ in their treatment of trend and seasonality.

ARIMA models provide more sophisticated methods for modeling trend and seasonal components than do exponential smoothing models, and, in particular, they allow the added benefit of including independent (predictor) variables in the model. This involves explicitly specifying autoregressive and moving average orders as well as the degree of differencing. You can include predictor variables and define transfer functions for any or all of them, as well as specify automatic detection of outliers or an explicit set of outliers.

Note: In practical terms, ARIMA models are most useful if you want to include predictors that might help to explain the behavior of the series that is being forecast, such as the number of catalogs that are mailed or the number of hits to a company web page. Exponential smoothing models describe the behavior of the time series without attempting to understand why it behaves as it does. For example, a series that historically peaks every 12 months will probably continue to do so even if you don't know why.

An Expert Modeler option is also available, which attempts to automatically identify and estimate the best-fitting ARIMA or exponential smoothing model for one or more target variables, thus eliminating the need to identify an appropriate model through trial and error. If in doubt, use the Expert Modeler option.

If predictor variables are specified, the Expert Modeler selects those variables that have a statistically significant relationship with the dependent series for inclusion in ARIMA models. Model variables are transformed where appropriate using differencing and/or a square root or natural log transformation. By default, the Expert Modeler considers all exponential smoothing models and all ARIMA models and picks the best model among them for each target field. You can, however, limit the Expert Modeler only to pick the best of the exponential smoothing models or only to pick the best of the ARIMA models. You can also specify automatic detection of outliers.

- [Time Series node - field options](#)
- [Time Series node - data specification options](#)
- [Time Series node - build options](#)
- [Time Series node - model options](#)
- [Time Series model nugget](#)

Time Series node - field options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign targets, predictors and other roles, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Targets. Select one or more of the fields as your target for the prediction.

Candidate inputs. Select one or more fields as inputs for the prediction. You can select only fields with a measurement level of continuous.

Events and Interventions. Use this area to designate certain input fields as event or intervention fields. This designation identifies a field as containing time series data that can be affected by events (predictable recurring situations; for example, sales promotions) or interventions (one-time incidents; for example, power outage or employee strike). The fields you select must be flags that have integer storage.

Time Series node - data specification options

The Data Specifications tab is where you set all the options for the data to be included in your model. As long as you specify both a Date/time field and Time interval, you can click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains several different panes on which you set the customizations that are specific to your model.

- [Time Series node - observations](#)
- [Time Series node - time interval for analysis](#)
- [Time Series node - aggregation and distribution options](#)
- [Time Series node - missing value options](#)
- [Time Series node - estimation period](#)

Time Series node - observations

Use the settings in this pane to specify the fields that define the observations.

Observations that are specified by a date/time field

You can specify that the observations are defined by a date, time, or timestamp field. In addition to the field that defines the observations, select the appropriate time interval that describes the observations. Depending on the specified time interval, you can also specify other settings, such as the interval between observations (increment) or the number of days per week. The following considerations apply to the time interval:

- Use the value Irregular when the observations are irregularly spaced in time, such as the time at which a sales order is processed. When Irregular is selected, you must specify the time interval that is used for the analysis, from the Time Interval settings on the Data Specifications tab.
- When the observations represent a date and time and the time interval is hours, minutes, or seconds, then use Hours per day, Minutes per day, or Seconds per day. When the observations represent a time (duration) without reference to a date and the time interval is hours, minutes, or seconds, then use Hours (non-periodic), Minutes (non-periodic), or Seconds (non-periodic).
- Based on the selected time interval, the procedure can detect missing observations. Detecting missing observations is necessary since the procedure assumes that all observations are equally spaced in time and that no observations are missing. For example, if the time interval is Days and the date 2015-10-27 is followed by 2015-10-29, then an observation is missing for 2015-10-28. Values are imputed for any missing observations; use the Missing Value Handling area of the Data Specifications tab to specify settings for handling missing values.
- The specified time interval allows the procedure to detect multiple observations in the same time interval that need to be aggregated together and to align observations on an interval boundary, such as the first of the month, to ensure that the observations are equally spaced. For example, if the time interval is Months, then multiple dates in the same month are aggregated together. This type of aggregation is referred to as *grouping*. By default, observations are summed when grouped. You can specify a different method for grouping, such as the mean of the observations, from the Aggregation and Distribution settings on the Data Specifications tab.

- For some time intervals, the additional settings can define breaks in the normal equally spaced intervals. For example, if the time interval is Days, but only weekdays are valid, you can specify that there are five days in a week, and the week begins on Monday.

Observations that are defined as periods or cyclic periods

Observations can be defined by one or more integer fields that represent periods or repeating cycles of periods, up to an arbitrary number of cycle levels. With this structure, you can describe series of observations that don't fit one of the standard time intervals. For example, a fiscal year with only 10 months can be described with a cycle field that represents years and a period field that represents months, where the length of one cycle is 10.

Fields that specify cyclic periods define a hierarchy of periodic levels, where the lowest level is defined by the Period field. The next highest level is specified by a cycle field whose level is 1, followed by a cycle field whose level is 2, and so on. Field values for each level, except the highest, must be periodic with respect to the next highest level. Values for the highest level cannot be periodic. For example, in the case of the 10-month fiscal year, months are periodic within years and years are not periodic.

- The length of a cycle at a particular level is the periodicity of the next lowest level. For the fiscal year example, there is only one cycle level and the cycle length is 10 since the next lowest level represents months and there are 10 months in the specified fiscal year.
- Specify the starting value for any periodic field that does not start from 1. This setting is necessary for detecting missing values. For example, if a periodic field starts from 2 but the starting value is specified as 1, then the procedure assumes that there is a missing value for the first period in each cycle of that field.

Time Series node - time interval for analysis

The time interval that you use for analysis can differ from the time interval of the observations. For example, if the time interval of the observations is Days, you might choose Months for the time interval for analysis. The data is then aggregated from daily to monthly data before the model is built. You can also choose to distribute the data from a longer to a shorter time interval. For example, if the observations are quarterly then you can distribute the data from quarterly to monthly data.

Use the settings in this pane to specify the time interval for the analysis. The method by which the data is aggregated or distributed is specified from the Aggregation and Distribution settings on the Data Specifications tab.

The available choices for the time interval at which the analysis is done depend on how the observations are defined and the time interval of those observations. In particular, when the observations are defined by cyclic periods, only aggregation is supported. In that case, the time interval of the analysis must be greater than or equal to the time interval of the observations.

Time Series node - aggregation and distribution options

Use the settings in this pane to specify settings for aggregating or distributing the input data with respect to the time intervals of the observations.

Aggregation functions

When the time interval that is used for the analysis is longer than the time interval of the observations, the input data are aggregated. For example, aggregation is done when the time interval of the observations is Days and the time interval for analysis is Months. The following aggregation functions are available: mean, sum, mode, min, or max.

Distribution functions

When the time interval that is used for the analysis is shorter than the time interval of the observations, the input data are distributed. For example, distribution is done when the time interval of the observations is Quarters and the time interval for analysis is Months. The following distribution functions are available: mean or sum.

Grouping functions

Grouping is applied when observations are defined by date/times and multiple observations occur in the same time interval. For example, if the time interval of the observations is Months, then multiple dates in the same month are grouped and associated with the month in which they occur. The following grouping functions are available: mean, sum, mode, min, or max. Grouping is always done when the observations are defined by date/times and the time interval of the observations is specified as Irregular.

Note: Although grouping is a form of aggregation, it is done before any handling of missing values whereas formal aggregation is done after any missing values are handled. When the time interval of the observations is specified as Irregular, aggregation is done only with the grouping function.

Aggregate cross-day observations to previous day

Specifies whether observations with times that cross a day boundary are aggregated to the values for the previous day. For example, for hourly observations with an eight-hour day that starts at 20:00, this setting specifies whether observations between 00:00 and 04:00 are included in the aggregated results for the previous day. This setting applies only if the time interval of the observations is Hours per day, Minutes per day or Seconds per day and the time interval for analysis is Days.

Custom settings for specified fields

You can specify aggregation, distribution, and grouping functions on a field by field basis. These settings override the default settings for the aggregation, distribution, and grouping functions.

Time Series node - missing value options

Use the settings in this pane to specify how any missing values in the input data are to be replaced with an imputed value. The following replacement methods are available:

Linear interpolation

Replaces missing values by using a linear interpolation. The last valid value before the missing value and the first valid value after the missing value are used for the interpolation. If the first or last observation in the series has a missing value, then the two nearest non-missing values at the beginning or end of the series are used.

Replaces missing values by using a linear interpolation.

- For non-seasonal data, the last valid value before the missing value and the first valid value after the missing value are used for the interpolation. If missing values are at the beginning or the end of a time series, then a linear extrapolation method is used based on the two nearest valid values.
- For seasonal data, a missing value is linearly interpolated using the last valid value of the same period before the missing value and the first valid value of the same period after the missing value. If one of the two values of the same period can't be found for the missing value, then the data will be regarded as non-seasonal data and linear interpolation of non-seasonal data is used to impute the missing value.

Series mean

Replaces missing values with the mean for the entire series.

Mean of nearby points

Replaces missing values with the mean of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the mean.

Median of nearby points

Replaces missing values with the median of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the median.

Linear trend

This option uses all non-missing observations in the series to fit a simple linear regression model, which is then used to impute the missing values.

Other settings:

Lowest data quality score (%)

Computes data quality measures for the time variable and for input data corresponding to each time series. If the data quality score is lower than this threshold, the corresponding time series will be discarded.

Time Series node - estimation period

In the Estimation Period pane, you can specify the range of records to be used in model estimation. By default, the estimation period starts at the time of the earliest observation and ends at the time of the latest observation across all series.

By start and end times

You can specify both the start and end of the estimation period or you can specify just the start or just the end. If you omit the start or the end of the estimation period, the default value is used.

- If the observations are defined by a date/time field, enter values for start and end in the same format that is used for the date/time field.
- For observations that are defined by cyclic periods, specify a value for each of the cyclic periods fields. Each field is displayed in a separate column.

By latest or earliest time intervals

Defines the estimation period as a specified number of time intervals that start at the earliest time interval or end at the latest time interval in the data, with an optional offset. In this context, the time interval refers to the time interval of the analysis. For example, assume that the observations are monthly but the time interval of the analysis is quarters. Specifying Latest and a value of 24 for the Number of time intervals means the latest 24 quarters.

Optionally, you can exclude a specified number of time intervals. For example, specifying the latest 24 time intervals and 1 for the number to exclude means that the estimation period consists of the 24 intervals that precede the last one.

Time Series node - build options

The Build Options tab is where you set all the options for building your model. You can, of course, just click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains two different panes on which you set the customizations that are specific to your model.

- [Time Series node - general build options](#)
- [Time Series node - build output options](#)

Time Series node - general build options

The options available on this pane depend on which of the following three settings you choose from the Method list:

- Expert Modeler. Choose this option to use the Expert Modeler, which automatically finds the best-fitting model for each dependent series.
- Exponential Smoothing. Use this option to specify a custom exponential smoothing model.
- ARIMA. Use this option to specify a custom ARIMA model.

Expert Modeler

Under Model Type, select the type of models you want to build:

- All models. The Expert Modeler considers both ARIMA and exponential smoothing models.
- Exponential smoothing models only. The Expert Modeler considers only exponential smoothing models.
- ARIMA models only. The Expert Modeler considers only ARIMA models.

Expert Modeler considers seasonal models. This option is only enabled if a periodicity is defined for the active dataset. When this option is selected, the Expert Modeler considers both seasonal and nonseasonal models. If this option is not selected, the Expert Modeler considers only nonseasonal models.

Expert Modeler considers sophisticated exponential smoothing models. When this option is selected, the Expert Modeler searches a total of 13 exponential smoothing models (7 of them existed in the original Time Series node, and 6 of them were added in version 18.1). If this option is not selected, the Expert Modeler only searches the original 7 exponential smoothing models.

Under Outliers, select from the following options

Detect outliers automatically. By default, automatic detection of outliers is not performed. Select this option to perform automatic detection of outliers, then select the desired outlier types.

For more information, see [Outliers](#).

Input fields must have a measurement level of *Flag*, *Nominal*, or *Ordinal* and must be numeric (for example, 1/0, not True/False, for a flag field), before they are included in this list.

For more information, see [Pulses and steps](#).

The Expert Modeler considers only simple regression and not arbitrary transfer functions for inputs that are identified as event or intervention fields on the Fields tab.

Exponential Smoothing

Model Type. Exponential smoothing models are classified as either seasonal or nonseasonal.¹ Seasonal models are only available if the periodicity defined by using the Time Intervals pane on the Data Specifications tab is seasonal. The seasonal periodicities are: cyclic periods, years, quarters, months, days per week, hours per day, minutes per day, and seconds per day. The following model types are available:

- Simple. This model is appropriate for a series in which there is no trend or seasonality. Its only relevant smoothing parameter is level. Simple exponential smoothing is most similar to an ARIMA with zero orders of autoregression, one order of differencing, one order of moving average, and no constant.
- Holt's linear trend. This model is appropriate for a series in which there is a linear trend and no seasonality. Its relevant smoothing parameters are level and trend, and, in this model, they are not constrained by each other's values. Holt's model is more general than Brown's model but may take longer to compute estimates for large series. Holt's exponential smoothing is most similar to an ARIMA with zero orders of autoregression, two orders of differencing, and two orders of moving average.
- Damped trend. This model is appropriate for a series in which there is a linear trend that is dying out and no seasonality. Its relevant smoothing parameters are level, trend, and damping trend. Damped exponential smoothing is most similar to an ARIMA with one order of autoregression, one order of differencing, and two orders of moving average.
- Multiplicative trend. This model is appropriate for a series in which there is a trend that changes with the magnitude of the series and no seasonality. Its relevant smoothing parameters are level and trend. Multiplicative trend exponential smoothing is not similar to any ARIMA model.
- Brown's linear trend. This model is appropriate for a series in which there is a linear trend and no seasonality. Its relevant smoothing parameters are level and trend, but, in this model, they are assumed to be equal. Brown's model is therefore a special case of Holt's

model. Brown's exponential smoothing is most similar to an ARIMA with zero orders of autoregression, two orders of differencing, and two orders of moving average, with the coefficient for the second order of moving average equal to one half of the coefficient for the first order squared.

- Simple seasonal. This model is appropriate for a series in which there is no trend and a seasonal effect that is constant over time. Its relevant smoothing parameters are level and season. Seasonal exponential smoothing is most similar to an ARIMA with zero orders of autoregression; one order of differencing; one order of seasonal differencing; and orders 1, p , and $p+1$ of moving average, where p is the number of periods in a seasonal interval. For monthly data, $p = 12$.
- Winters' additive. This model is appropriate for a series in which there is a linear trend and a seasonal effect that is constant over time. Its relevant smoothing parameters are level, trend, and season. Winters' additive exponential smoothing is most similar to an ARIMA with zero orders of autoregression; one order of differencing; one order of seasonal differencing; and $p+1$ orders of moving average, where p is the number of periods in a seasonal interval. For monthly data, $p=12$.
- Damped trend with additive seasonal. This model is appropriate for a series in which there is a linear trend that is dying out and a seasonal effect that is constant over time. Its relevant smoothing parameters are level, trend, damping trend, and season. Damped trend and additive seasonal exponential smoothing is not similar to any ARIMA model.
- Multiplicative trend with additive seasonal. This model is appropriate for a series in which there is a trend that changes with the magnitude of the series and a seasonal effect that is constant over time. Its relevant smoothing parameters are level, trend, and season. Multiplicative trend and additive seasonal exponential smoothing is not similar to any ARIMA model.
- Multiplicative seasonal. This model is appropriate for a series in which there is no trend and a seasonal effect that changes with the magnitude of the series. Its relevant smoothing parameters are level and season. Multiplicative seasonal exponential smoothing is not similar to any ARIMA model.
- Winters' multiplicative. This model is appropriate for a series in which there is a linear trend and a seasonal effect that changes with the magnitude of the series. Its relevant smoothing parameters are level, trend, and season. Winters' multiplicative exponential smoothing is not similar to any ARIMA model.
- Damped trend with multiplicative seasonal. This model is appropriate for a series in which there is a linear trend that is dying out and a seasonal effect that changes with the magnitude of the series. Its relevant smoothing parameters are level, trend, damping trend, and season. Damped trend and multiplicative seasonal exponential smoothing is not similar to any ARIMA model.
- Multiplicative trend with multiplicative seasonal. This model is appropriate for a series in which there are a trend and a seasonal effect that both change with the magnitude of the series. Its relevant smoothing parameters are level, trend, and season. Multiplicative trend and multiplicative seasonal exponential smoothing is not similar to any ARIMA model.

Target Transformation. You can specify a transformation to be performed on each dependent variable before it is modeled.

For more information, see [Series transformations](#).

- None. No transformation is performed.
- Square root. Square root transformation is performed.
- Natural log. Natural log transformation is performed.

ARIMA

Specify the structure of a custom ARIMA model.

ARIMA Orders. Enter values for the various ARIMA components of your model into the corresponding cells of the grid. All values must be non-negative integers. For autoregressive and moving average components, the value represents the maximum order. All positive lower orders are included in the model. For example, if you specify 2, the model includes orders 2 and 1. Cells in the Seasonal column are only enabled if a periodicity is defined for the active dataset.

- Autoregressive (p). The number of autoregressive orders in the model. Autoregressive orders specify which previous values from the series are used to predict current values. For example, an autoregressive order of 2 specifies that the value of the series two time periods in the past is used to predict the current value.
- Difference (d). Specifies the order of differencing applied to the series before estimating models. Differencing is necessary when trends are present (series with trends are typically nonstationary and ARIMA modeling assumes stationarity) and is used to remove their effect. The order of differencing corresponds to the degree of series trend; first-order differencing accounts for linear trends, second-order differencing accounts for quadratic trends, and so on.
- Moving Average (q). The number of moving average orders in the model. Moving average orders specify how deviations from the series mean for previous values are used to predict current values. For example, moving-average orders of 1 and 2 specify that deviations from the mean value of the series from each of the last two time periods be considered when predicting current values of the series.

Seasonal. Seasonal autoregressive, moving average, and differencing components play the same roles as their nonseasonal counterparts. For seasonal orders, however, current series values are affected by previous series values that are separated by one or more seasonal periods. For example, for monthly data (seasonal period of 12), a seasonal order of 1 means that the current series value is affected by the series value 12 periods before the current one. A seasonal order of 1, for monthly data, is then the same as specifying a nonseasonal order of 12.

Detect outliers automatically. Select this option to perform automatic detection of outliers, and select one or more of the outlier types available.

Type of Outliers to Detect. Select the outlier type(s) you want to detect. The supported types are:

- Additive (default)
- Level shift (default)
- Innovational

- Transient
- Seasonal additive
- Local trend
- Additive patch

Transfer Function Orders and Transformations. To specify transformations and to define transfer functions for any or all of the input fields in your ARIMA model, click Set; a separate dialog box is displayed in which you enter the transfer and transformation details.

Include constant in model. Inclusion of a constant is standard unless you are sure that the overall mean series value is 0. Excluding the constant is recommended when differencing is applied.

Further details

- For more information on types of outliers, see [Outliers](#).
- For more information on transfer and transformation functions, see [Transfer and transformation functions](#).
- [Transfer and transformation functions](#)

¹ Gardner, E. S. 1985. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4, 1-28.

Transfer and transformation functions

Use the Transfer Function Orders and Transformations dialog box to specify transformations and to define transfer functions for any or all of the input fields in your ARIMA model.

Target Transformations. In this pane you can specify a transformation to be performed on each target variable before it is modeled.

- None. No transformation is performed.
- Square root. Square root transformation is performed.
- Natural log. Natural log transformation is performed.

For more information, see [Series transformations](#).

Candidate Inputs Transfer functions and Transformation. You use transfer functions to specify the manner in which past values of the input fields are used to forecast future values of the target series. The list on the left hand side of the pane shows all input fields. The remaining information in this pane is specific to the input field you select.

Transfer Function Orders. Enter values for the various components of the transfer function into the corresponding cells of the Structure grid. All values must be non-negative integers. For numerator and denominator components, the value represents the maximum order. All positive lower orders are included in the model. In addition, order 0 is always included for numerator components. For example, if you specify 2 for numerator, the model includes orders 2, 1, and 0. If you specify 3 for denominator, the model includes orders 3, 2, and 1. Cells in the Seasonal column are only enabled if a periodicity is defined for the active dataset.

Numerator. The numerator order of the transfer function specifies which previous values from the selected independent (predictor) series are used to predict current values of the dependent series. For example, a numerator order of 1 specifies that the value of an independent series one time period in the past, in addition to the current value of the independent series, is used to predict the current value of each dependent series.

Denominator. The denominator order of the transfer function specifies how deviations from the series mean, for previous values of the selected independent (predictor) series, are used to predict current values of the dependent series. For example, a denominator order of 1 specifies that deviations from the mean value of an independent series one time period in the past is considered when predicting the current value of each dependent series.

Difference. Specifies the order of differencing applied to the selected independent (predictor) series before estimating models. Differencing is necessary when trends are present and is used to remove their effect.

Seasonal. Seasonal numerator, denominator, and differencing components play the same roles as their nonseasonal counterparts. For seasonal orders, however, current series values are affected by previous series values that are separated by one or more seasonal periods. For example, for monthly data (seasonal period of 12), a seasonal order of 1 means that the current series value is affected by the series value 12 periods prior to the current one. A seasonal order of 1, for monthly data, is then the same as specifying a nonseasonal order of 12.

Delay. Setting a delay causes the input field's influence to be delayed by the number of intervals specified. For example, if the delay is set to 5, the value of the input field at time t doesn't affect forecasts until five periods have elapsed ($t + 5$).

Transformation. Specification of a transfer function for a set of independent variables also includes an optional transformation to be performed on those variables.

- None. No transformation is performed.
- Square root. Square root transformation is performed.
- Natural log. Natural log transformation is performed.

Time Series node - build output options

Maximum number of lags in ACF and PACF output. Autocorrelation (ACF) and partial autocorrelation (PACF) are measures of association between current and past series values and indicate which past series values are most useful in predicting future values. You can set the maximum number of lags that are shown in tables and plots of autocorrelations and partial autocorrelations.

For more information, see [Autocorrelation and partial autocorrelation functions](#).

Calculate predictor importance. For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring the predictors that matter least. Predictor importance can take longer to calculate for some models, particularly when you are working with large datasets, and is off by default for some models as a result.

For more information, see [Predictor Importance](#).

Time Series node - model options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Confidence limit width (%). Confidence intervals are computed for the model predictions and residual autocorrelations. You can specify any positive value less than 100. By default, a 95% confidence interval is used.

Continue estimation using existing model(s). If you already generated a time series model, select this option to reuse the criteria settings that are specified for that model and generate a new model node in the Models palette, rather than building a new model from the beginning. In this way, you can save time by re-estimating and producing a new forecast that is based on the same model settings as before but using more recent data. Thus, for example, if the original model for a particular time series was Holt's linear trend, the same type of model is used for reestimating and forecasting for that data. The system does not reattempt to find the best model type for the new data.

Build scoring model only. To reduce the amount of data that is stored in the model, check this box. Using this option can improve performance when building models with large numbers of time series (tens of thousands). You can still score the data in the usual way.

Extend records into the future. Enables the following Future Values to Use in Forecasting section, where you can set the number of time intervals to forecast beyond the end of the estimation period. The time interval in this case is the time interval of the analysis, which you specify on the Data Specifications tab. There is no maximum limit for this setting. Using the following options, you can automatically compute future values of inputs, or you can manually specify forecasting values for one or more predictors.

Future Values to Use in Forecasting

- Compute future values of inputs If you select this option, the forecast values for predictors, noise predictions, variance estimation, and future time values are calculated automatically. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period.
 - Select fields whose values you wish to add to the data. For each record that you want to forecast (excluding holdouts), if you are using predictor fields (with the role set to **Input**), you can specify estimated values for the forecast period for each predictor. You can either specify values manually, or choose from a list.
 - Field. Click the field selector button and choose any fields that may be used as predictors. Note that fields selected here may or may not be used in modeling; to actually use a field as a predictor, it must be selected in a downstream modeling node. This dialog box simply gives you a convenient place to specify future values so they can be shared by multiple downstream modeling nodes without specifying them separately in each node. Also note that the list of available fields may be constrained by selections on the Build Options tab.
Note that if future values are specified for a field that is no longer available in the stream (because it has been dropped or because of updated selections made on the Build Options tab), the field is shown in red.
 - Values. For each field, you can choose from a list of functions, or click Specify to either enter values manually or choose from a list of predefined values. If the predictor fields relate to items that are under your control, or which are otherwise knowable in advance, you should enter values manually. For example, if you are forecasting next month's revenues for a hotel based on the number of room reservations, you could specify the number of reservations you actually have for that period. Conversely, if a predictor field relates to something outside your control, such as a stock price, you could use a function such as the most recent value or the mean of recent points.
- The available functions depend on the measurement level of the field.

Table 1. Functions available for measurement levels

Measurement level	Functions
-------------------	-----------

Measurement level	Functions
Continuous or Nominal field	Blank Mean of recent points Most recent value Specify
Flag field	Blank Most recent value True False Specify

Mean of recent points calculates the future value from the mean of the last three data points.

Most recent value sets the future value to that of the most recent data point.

True/False sets the future value of a flag field to True or False as specified.

Specify opens a dialog box for specifying future values manually, or choosing them from a predefined list.

Make Available for Scoring

You can set the default values here for the scoring options that appear on the dialog box for the model nugget.

- Calculate upper and lower confidence limits. If selected, this option creates new fields (with the default prefixes \$TSLCI- and \$TSUCI-) for the lower and upper confidence intervals, for each target field.
- Calculate noise residuals. If selected, this option creates a new field (with the default prefix \$TSResidual-) for the model residuals for each target field, together with a total of these values.

Model Settings

Maximum number of models to be displayed in output. Specify the maximum number of models you want to include in the output. Note that if the number of models built exceeds this threshold, the models are not shown in the output but they're still available for scoring. Default value is 10. Displaying a large number of models may result in poor performance or instability.

Time Series model nugget

- [Time Series model nugget output](#)
- [Time Series model nugget settings](#)

Time Series model nugget output

After you create a Time Series model, the following information is available in the Output viewer. Note that there's a limit of 10 models that can be displayed in the Output viewer for Time Series models.

Temporal Information Summary

The summary shows the following information:

- The Time field
- The increment
- The start and end point
- The number of unique points

The summary applies to all targets.

Model information table

Repeated for each target, the Model information table provides key information about the model. The table always includes the following high-level model settings:

- The name of the target field that is selected in either the Type node or the Time Series node Fields tab.
- The model building method - for example, Exponential Smoothing, or ARIMA.
- The number of predictors input into the model.

- The number of records that were used to fit the model type. Examples of the different types of models might include: RMSE, MAE, AIC, BIC, and R Square.

In addition, the Ljung-Box Q statistic might also be shown if your data meets the required conditions. This statistic is **not** available under the following conditions:

- If the number of non-missing data points is less than or equal to the number of summation terms desired (fixed at 18).
- If the number of parameters is greater than or equal to the number of summation terms desired.
- If the number of summation terms that are computed is less than the smallest acceptable k value (fixed at 7).
- If the table repeats for each target.

Predictor Importance

Repeated for each target, the Predictor Importance graph shows the importance of the top 10 inputs (predictors) in the model as a bar chart.

If there are more than 10 fields in the chart, you can change the selection of predictors that are included in the chart by using the slider beneath the chart. The indicator marks on the slider are a fixed width, and each mark on the slider represents 10 fields. You can move the indicator marks along the slider to display the next or previous 10 fields, ordered by predictor importance.

You can double-click the chart to open a separate dialog box in which you can edit the graph settings. For example, you can amend items such as the size of the graph, and the size and color of the fonts used. When you close this separate editing dialog box, the changes are applied to the chart that is displayed in the Output tab.

Correlogram

A correlogram, or autocorrelation plot, is shown for each target and shows the autocorrelation function (ACF), or partial autocorrelation function (PACF), of the residuals (the differences between expected and actual values) versus the time lags. The confidence interval is shown as a highlight across the chart.

Parameter Estimates

Repeated for each target, the Parameter Estimates table shows (where applicable) the following details:

- Target name
- The transformation applied
- The lags used for this parameter in the model (ARIMA)
- The coefficient value
- The standard error of the parameter estimate
- The value of the parameter estimate divided by the standard error
- The significance level for the parameter estimate.

Time Series model nugget settings

The Settings tab provides additional options for the Time Series model nugget.

Forecast

The option to Extend records into the future. sets the number of time intervals to forecast beyond the end of the estimation period. The time interval in this case is the time interval of the analysis, which is specified on the Data Specifications tab of the Time Series node. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period.

Compute future values of inputs. If you select this option, the forecast values for predictors, noise predictions, variance estimation, and future time values are calculated.

Future Values to Use in Forecasting

- Compute future values of inputs If you select this option, the forecast values for predictors, noise predictions, variance estimation, and future time values are calculated automatically. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period.
- Select fields whose values you wish to add to the data. For each record that you want to forecast (excluding holdouts), if you are using predictor fields (with the role set to **Input**), you can specify estimated values for the forecast period for each predictor. You can either specify values manually, or choose from a list.

- Field. Click the field selector button and choose any fields that may be used as predictors. Note that fields selected here may or may not be used in modeling; to actually use a field as a predictor, it must be selected in a downstream modeling node. This dialog box simply gives you a convenient place to specify future values so they can be shared by multiple downstream modeling nodes without specifying them separately in each node. Also note that the list of available fields may be constrained by selections on the Build Options tab.
Note that if future values are specified for a field that is no longer available in the stream (because it has been dropped or because of updated selections made on the Build Options tab), the field is shown in red.
- Values. For each field, you can choose from a list of functions, or click Specify to either enter values manually or choose from a list of predefined values. If the predictor fields relate to items that are under your control, or which are otherwise knowable in advance, you should enter values manually. For example, if you are forecasting next month's revenues for a hotel based on the number of room reservations, you could specify the number of reservations you actually have for that period. Conversely, if a predictor field relates to something outside your control, such as a stock price, you could use a function such as the most recent value or the mean of recent points.
The available functions depend on the measurement level of the field.

Table 1. Functions available for measurement levels

Measurement level	Functions
Continuous or Nominal field	Blank Mean of recent points Most recent value Specify
Flag field	Blank Most recent value True False Specify

Mean of recent points calculates the future value from the mean of the last three data points.

Most recent value sets the future value to that of the most recent data point.

True/False sets the future value of a flag field to True or False as specified.

Specify opens a dialog box for specifying future values manually, or choosing them from a predefined list.

Make available for scoring

Create new fields for each model to be scored. Enables you to specify the new fields to create for each model to be scored.

- Noises Residuals. If selected, creates a new field (with the default prefix \$TSResidual-) for the model residuals for each target field, together with a total of these values.
- Upper and lower confidence limits. If selected, this option creates new fields (with the default prefixes \$TSLCI- and \$TSUCI-) for the lower and upper confidence intervals respectively, for each target field, together with totals of these values.

Targets included for scoring. Select available targets to include in the model score.

Self-Learning Response Node Models

- [SLRM node](#)
- [SLRM Model Nuggets](#)

SLRM node

The Self-Learning Response Model (SLRM) node enables you to build a model that you can continually update, or reestimate, as a dataset grows without having to rebuild the model every time using the complete dataset. For example, this is useful when you have several products and you want to identify which product a customer is most likely to buy if you offer it to them. This model allows you to predict which offers are most appropriate for customers and the probability of their acceptance.

The model can initially be built using a small dataset with randomly made offers and the responses to those offers. As the dataset grows, the model can be updated and therefore becomes more able to predict the most suitable offers for customers and the probability of their acceptance based upon other input fields such as age, gender, job, and income. The offers available can be changed by adding or removing them from within the node dialog box, instead of having to change the target field of the dataset.

When coupled with IBM® SPSS® Collaboration and Deployment Services, you can set up automatic regular updates to the model. This process, without the need for human oversight or action, provides a flexible and low-cost solution for organizations and applications where custom intervention by a data miner is not possible or necessary.

Example. A financial institution wants to achieve more profitable results by matching the offer that is most likely to be accepted to each customer. You can use a self-learning model to identify the characteristics of customers most likely to respond favorably based on previous promotions and to update the model in real time based on the latest customer responses.

- [SLRM Node Fields Options](#)
 - [SLRM Node Model Options](#)
 - [SLRM Node Settings Options](#)
-

SLRM Node Fields Options

Before executing an SLRM node, you must specify both the target and target response fields on the Fields tab of the node.

Target field. Select the target field from the list; for example, a nominal (set) field containing the different products you want to offer to customers.

Note: The target field must have string storage, not numeric.

Target response field. Select the target response field from the list. For example, Accepted or Rejected.

Note: This field must be a Flag. The true value of the flag indicates offer acceptance and the false value indicates offer refusal.

The remaining fields in this dialog box are the standard ones used throughout IBM® SPSS® Modeler. See the topic [Modeling Node Fields Options](#) for more information.

Note: If the source data includes ranges that are to be used as continuous (numeric range) input fields, you must ensure that the metadata includes both the minimum and maximum details for each range.

Related information

- [SLRM node](#)
 - [SLRM Node Model Options](#)
 - [SLRM Node Settings Options](#)
 - [SLRM Model Nuggets](#)
 - [SLRM Model Settings](#)
-

SLRM Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Continue training existing model. By default, a completely new model is created each time a modeling node is executed. If this option is selected, training continues with the last model successfully produced by the node. This makes it possible to update or refresh an existing model without having to access the original data and may result in significantly faster performance since *only* the new or updated records are fed into the stream. Details on the previous model are stored with the modeling node, making it possible to use this option even if the previous model nugget is no longer available in the stream or Models palette.

Target field values By default this is set to Use all, which means that a model will be built that contains every offer associated with the selected target field value. If you want to generate a model that contains only some of the target field's offers, click Specify and use the Add, Edit, and Delete buttons to enter or amend the names of the offers for which you want to build a model. For example, if you chose a target that lists all of the products you supply, you can use this field to limit the offered products to just a few that you enter here.

Model Assessment. The fields in this panel are independent from the model in that they don't affect the scoring. Instead they enable you to create a visual representation of how well the model will predict results.

Note: To display the model assessment results in the model nugget you must also select the Display model evaluation box.

- **Include model assessment.** Select this box to create graphs that show the model's predicted accuracy for each selected offer.
- **Set random seed.** When estimating the accuracy of a model based on a random percentage, this option allows you to duplicate the same results in another session. By specifying the starting value used by the random number generator, you can ensure the same records are

assigned each time the node is executed. Enter the desired seed value. If this option is not selected, a different sample will be generated each time the node is executed.

- **Simulated sample size.** Specify the number of records to be used in the sample when assessing the model. The default is 100.
- **Number of iterations.** This enables you to stop building the model assessment after the number of iterations specified. Specify the maximum number of iterations; the default is 20.

Note: Bear in mind that large sample sizes and high numbers of iterations will increase the amount of time it takes to build the model.

Display model evaluation. Select this option to display a graphical representation of the results in the model nugget.

Related information

- [SLRM node](#)
 - [SLRM Node Fields Options](#)
 - [SLRM Node Settings Options](#)
 - [SLRM Model Nuggets](#)
 - [SLRM Model Settings](#)
-

SLRM Node Settings Options

The node settings options allow you to fine-tune the model-building process.

Maximum number of predictions per record. This option allows you to limit the number of predictions made for each record in the dataset. The default is 3.

For example, you may have six offers (such as savings, mortgage, car loan, pension, credit card, and insurance), but you only want to know the best two to recommend; in this case you would set this field to 2. When you build the model and attach it to a table, you would see two prediction columns (and the associated confidence in the probability of the offer being accepted) per record. The predictions could be made up of any of the six possible offers.

Level of randomization. To prevent any bias—for example, in a small or incomplete dataset—and treat all potential offers equally, you can add a level of randomization to the selection of offers and the probability of their being included as recommended offers. Randomization is expressed as a percentage, shown as decimal values between 0.0 (no randomization) and 1.0 (completely random). The default is 0.0.

Set random seed. When adding a level of randomization to selection of an offer, this option allows you to duplicate the same results in another session. By specifying the starting value used by the random number generator, you can ensure the same records are assigned each time the node is executed. Enter the desired seed value. If this option is not selected, a different sample will be generated each time the node is executed.

Note: When using the Set random seed option with records read from a database, a Sort node may be required prior to sampling in order to ensure the same result each time the node is executed. This is because the random seed depends on the order of records, which is not guaranteed to stay the same in a relational database.

Sort order. Select the order in which offers are to be displayed in the built model:

- Descending. The model displays offers with the highest scores first. These are the offers that have the greatest probability of being accepted.
- Ascending. The model displays offers with the lowest scores first. These are the offers that have the greatest probability of being rejected. For example, this may be useful when deciding which customers to remove from a marketing campaign for a specific offer.

Preferences for target fields. When building a model, there may be certain aspects of the data that you want to actively promote or remove. For example, if building a model that selects the best financial offer to promote to a customer, you may want to ensure that one particular offer is always included regardless of how well it scores against each customer.

To include an offer in this panel and edit its preferences, click Add, type the offer's name (for example, Savings or Mortgage), and click OK.

- Value. This shows the name of the offer that you added.
- Preference. Specify the level of preference to be applied to the offer. Preference is expressed as a percentage, shown as decimal values between 0.0 (not preferred) and 1.0 (most preferred). The default is 0.0.
- Always include. To ensure that a specific offer is always included in the predictions, select this box.
Note: If the Preference is set to 0.0, the Always include setting is ignored.

Take account of model reliability. A well-structured, data-rich model that has been fine-tuned through several regenerations should always produce more accurate results compared to a brand new model with little data. To take advantage of the more mature model's increased reliability, select this box.

Related information

- [SLRM node](#)
- [SLRM Node Fields Options](#)

- [SLRM Node Model Options](#)
 - [SLRM Model Nuggets](#)
 - [SLRM Model Settings](#)
-

SLRM Model Nuggets

Note: Results are only shown on this tab if you select both Include model assessment and Display model evaluation on the Model options tab.

When you run a stream containing an SLRM model, the node estimates the accuracy of the predictions for each target field value (offer) and the importance of each predictor used.

Note: If you selected Continue training existing model on the modeling node Model tab, the information shown on the model nugget is updated each time you regenerate the model.

For models built using IBM® SPSS® Modeler 12.0 or later, the model nugget Model tab is divided into two columns:

Left column.

- **View.** When you have more than one offer, select the one for which you want to display results.
- **Model Performance.** This shows the estimated model accuracy of each offer. The test set is generated through simulation.

Right column.

- **View.** Select whether you want to display Association with Response or Variable Importance details.
- **Association with Response.** Displays the association (correlation) of each predictor with the target variable.
- **Predictor Importance.** Indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least. This chart can be interpreted in the same manner as for other models that display predictor importance, though in the case of SLRM the graph is generated through simulation by the SLRM algorithm. This is done by removing each predictor in turn from the model and seeing how this affects the model's accuracy. See the topic [Predictor Importance](#) for more information.
- [SLRM Model Settings](#)

Related information

- [SLRM node](#)
 - [SLRM Node Fields Options](#)
 - [SLRM Node Model Options](#)
 - [SLRM Node Settings Options](#)
 - [SLRM Model Settings](#)
-

SLRM Model Settings

The Settings tab for a SLRM model nugget specifies options for modifying the built model. For example, you may use the SLRM node to build several different models using the same data and settings, then use this tab in each model to slightly modify the settings to see how that affects the results.

Note: This tab is only available after the model nugget has been added to a stream.

Maximum number of predictions per record. This option allows you to limit the number of predictions made for each record in the dataset. The default is 3.

For example, you may have six offers (such as savings, mortgage, car loan, pension, credit card, and insurance), but you only want to know the best two to recommend; in this case you would set this field to 2. When you build the model and attach it to a table, you would see two prediction columns (and the associated confidence in the probability of the offer being accepted) per record. The predictions could be made up of any of the six possible offers.

Level of randomization. To prevent any bias—for example, in a small or incomplete dataset—and treat all potential offers equally, you can add a level of randomization to the selection of offers and the probability of their being included as recommended offers. Randomization is expressed as a percentage, shown as decimal values between 0.0 (no randomization) and 1.0 (completely random). The default is 0.0.

Set random seed. When adding a level of randomization to selection of an offer, this option allows you to duplicate the same results in another session. By specifying the starting value used by the random number generator, you can ensure the same records are assigned each time the node is executed. Enter the desired seed value. If this option is not selected, a different sample will be generated each time the node is executed.

Note: When using the Set random seed option with records read from a database, a Sort node may be required prior to sampling in order to ensure the same result each time the node is executed. This is because the random seed depends on the order of records, which is not

guaranteed to stay the same in a relational database.

Sort order. Select the order in which offers are to be displayed in the built model:

- Descending. The model displays offers with the highest scores first. These are the offers that have the greatest probability of being accepted.
- Ascending. The model displays offers with the lowest scores first. These are the offers that have the greatest probability of being rejected. For example, this may be useful when deciding which customers to remove from a marketing campaign for a specific offer.

Preferences for target fields. When building a model, there may be certain aspects of the data that you want to actively promote or remove. For example, if building a model that selects the best financial offer to promote to a customer, you may want to ensure that one particular offer is always included regardless of how well it scores against each customer.

To include an offer in this panel and edit its preferences, click Add, type the offer's name (for example, Savings or Mortgage), and click OK.

- Value. This shows the name of the offer that you added.
- Preference. Specify the level of preference to be applied to the offer. Preference is expressed as a percentage, shown as decimal values between 0.0 (not preferred) and 1.0 (most preferred). The default is 0.0.
- Always include. To ensure that a specific offer is always included in the predictions, select this box.
Note: If the Preference is set to 0.0, the Always include setting is ignored.

Take account of model reliability. A well-structured, data-rich model that has been fine-tuned through several regenerations should always produce more accurate results compared to a brand new model with little data. To take advantage of the more mature model's increased reliability, select this box.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Related information

- [SLRM node](#)
- [SLRM Node Fields Options](#)
- [SLRM Node Model Options](#)
- [SLRM Node Settings Options](#)
- [SLRM Model Nuggets](#)

Support Vector Machine Models

- [About SVM](#)
- [How SVM Works](#)
- [Tuning an SVM Model](#)
- [SVM node](#)
- [SVM Model Nugget](#)
- [LSVM Node](#)
- [LSVM Model Nugget \(interactive output\)](#)

About SVM

Support Vector Machine (SVM) is a robust classification and regression technique that maximizes the predictive accuracy of a model without overfitting the training data. SVM is particularly suited to analyzing data with very large numbers (for example, thousands) of predictor fields.

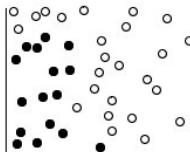
SVM has applications in many disciplines, including customer relationship management (CRM), facial and other image recognition, bioinformatics, text mining concept extraction, intrusion detection, protein structure prediction, and voice and speech recognition.

How SVM Works

SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane. Following this, characteristics of new data can be used to predict the group to which a new record should belong.

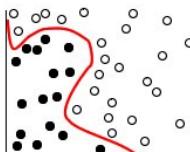
For example, consider the following figure, in which the data points fall into two different categories.

Figure 1. Original dataset



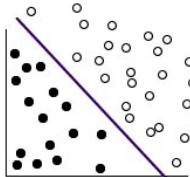
The two categories can be separated with a curve, as shown in the following figure.

Figure 2. Data with separator added



After the transformation, the boundary between the two categories can be defined by a hyperplane, as shown in the following figure.

Figure 3. Transformed data



The mathematical function used for the transformation is known as the **kernel** function. SVM in IBM® SPSS® Modeler supports the following kernel types:

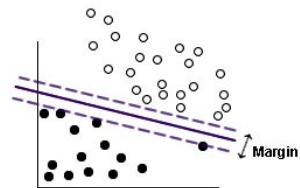
- Linear
- Polynomial
- Radial basis function (RBF)
- Sigmoid

A linear kernel function is recommended when linear separation of the data is straightforward. In other cases, one of the other functions should be used. You will need to experiment with the different functions to obtain the best model in each case, as they each use different algorithms and parameters.

Tuning an SVM Model

Besides the separating line between the categories, a classification SVM model also finds marginal lines that define the space between the two categories.

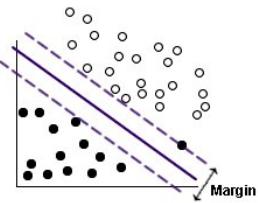
Figure 1. Data with a preliminary model



The data points that lie on the margins are known as the **support vectors**.

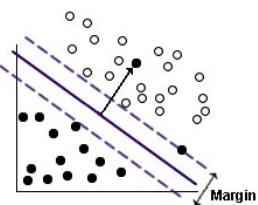
The wider the margin between the two categories, the better the model will be at predicting the category for new records. In the previous example, the margin is not very wide, and the model is said to be **overfitted**. A small amount of misclassification can be accepted in order to widen the margin; an example of this is shown in the following figure.

Figure 2. Data with an improved model



In some cases, linear separation is more difficult; an example of this is shown in the following figure.

Figure 3. A problem for linear separation



In a case like this, the goal is to find the optimum balance between a wide margin and a small number of misclassified data points. The kernel function has a **regularization parameter** (known as C) which controls the trade-off between these two values. You will probably need to experiment with different values of this and other kernel parameters in order to find the best model.

SVM node

The SVM node enables you to use a support vector machine to classify data. SVM is particularly suited for use with wide datasets, that is, those with a large number of predictor fields. You can use the default settings on the node to produce a basic model relatively quickly, or you can use the Expert settings to experiment with different types of SVM model.

When the model has been built, you can:

- Browse the model nugget to display the relative importance of the input fields in building the model.
- Append a Table node to the model nugget to view the model output.

Example. A medical researcher has obtained a dataset containing characteristics of a number of human cell samples extracted from patients who were believed to be at risk of developing cancer. Analysis of the original data showed that many of the characteristics differed significantly between benign and malignant samples. The researcher wants to develop an SVM model that can use the values of similar cell characteristics in samples from other patients to give an early indication of whether their samples might be benign or malignant.

- [SVM Node Model Options](#)
- [SVM Node Expert Options](#)

SVM Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Create split models. Builds a separate model for each possible value of input fields that are specified as split fields. See [Building Split Models](#) for more information.

Related information

- [About SVM](#)
- [How SVM Works](#)
- [Tuning an SVM Model](#)

SVM Node Expert Options

If you have detailed knowledge of support vector machines, expert options allow you to fine-tune the training process. To access the expert options, set Mode to Expert on the Expert tab.

Append all probabilities (valid only for categorical targets). If selected (checked), specifies that probabilities for each possible value of a nominal or flag target field are displayed for each record processed by the node. If this option is not selected, the probability of only the predicted value is displayed for nominal or flag target fields. The setting of this check box determines the default state of the corresponding check box on the model nugget display.

Stopping criteria. Determines when to stop the optimization algorithm. Values range from 1.0E-1 to 1.0E-6; default is 1.0E-3. Reducing the value results in a more accurate model, but the model will take longer to train.

Regularization parameter (C). Controls the trade-off between maximizing the margin and minimizing the training error term. Value should normally be between 1 and 10 inclusive; default is 10. Increasing the value improves the classification accuracy (or reduces the regression error) for the training data, but this can also lead to overfitting.

Regression precision (epsilon). Used only if the measurement level of the target field is *Continuous*. Causes errors to be accepted provided that they are less than the value specified here. Increasing the value may result in faster modeling, but at the expense of accuracy.

Kernel type. Determines the type of kernel function used for the transformation. Different kernel types cause the separator to be calculated in different ways, so it is advisable to experiment with the various options. Default is RBF (Radial Basis Function).

RBF gamma. Enabled only if the kernel type is set to RBF. Value should normally be between $3/k$ and $6/k$, where k is the number of input fields. For example, if there are 12 input fields, values between 0.25 and 0.5 would be worth trying. Increasing the value improves the classification accuracy (or reduces the regression error) for the training data, but this can also lead to overfitting.

Gamma. Enabled only if the kernel type is set to Polynomial or Sigmoid. Increasing the value improves the classification accuracy (or reduces the regression error) for the training data, but this can also lead to overfitting.

Bias. Enabled only if the kernel type is set to Polynomial or Sigmoid. Sets the `coef0` value in the kernel function. The default value 0 is suitable in most cases.

Degree. Enabled only if Kernel type is set to Polynomial. Controls the complexity (dimension) of the mapping space. Normally you would not use a value greater than 10.

Related information

- [About SVM](#)
 - [How SVM Works](#)
 - [Tuning an SVM Model](#)
-

SVM Model Nugget

The SVM model creates a number of new fields. The most important of these is the `$S-fieldname` field, which shows the target field value predicted by the model.

The number and names of the new fields created by the model depend on the measurement level of the target field (this field is indicated in the following tables by *fieldname*).

To see these fields and their values, add a Table node to the SVM model nugget and execute the Table node.

Table 1. Target field measurement level is 'Nominal' or 'Flag'

New field name	Description
<code>\$S-fieldname</code>	Predicted value of target field.
<code>\$SP-fieldname</code>	Probability of predicted value.
<code>\$SP-value</code>	Probability of each possible value of nominal or flag (displayed only if Append all probabilities is checked on the Settings tab of the model nugget).
<code>\$SRP-value</code>	(Flag targets only) Raw (SRP) and adjusted (SAP) propensity scores, indicating the likelihood of a "true" outcome for the target field. These scores are displayed only if the corresponding check boxes are selected on the Analyze tab of the SVM modeling node before the model is generated. See the topic Modeling Node Analyze Options for more information.
<code>\$SAP-value</code>	

Table 2. Target field measurement level is
'Continuous'

New field name	Description
<code>\$S-fieldname</code>	Predicted value of target field.

Predictor Importance

Optionally, a chart that indicates the relative importance of each predictor in estimating the model may also be displayed on the Model tab. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least. Note this chart is only available if Calculate predictor importance is selected on the Analyze tab before generating the model. See the topic [Predictor Importance](#) for more information.

Note: Predictor importance may take longer to calculate for SVM than for other types of models, and is not selected on the Analyze tab by default. Selecting this option may slow performance, particularly with large datasets.

- [SVM Model Settings](#)

SVM Model Settings

The Settings tab enables you to specify extra fields to be displayed when viewing the results (for example by executing a Table node attached to the nugget). You can see the effect of each of these options by selecting them and clicking the Preview button--scroll to the right of the Preview output to see the extra fields.

Append all probabilities (valid only for categorical targets). If this option is checked, probabilities for each possible value of a nominal or flag target field are displayed for each record processed by the node. If this option is unchecked, only the predicted value and its probability are displayed for nominal or flag target fields.

The default setting of this check box is determined by the corresponding check box on the modeling node.

Calculate raw propensity scores. For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

Calculate adjusted propensity scores. Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score outside of the Database If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

LSVM Node

The LSVM node enables you to use a linear support vector machine to classify data. LSVM is particularly suited for use with wide datasets, that is, those with a large number of predictor fields. You can use the default settings on the node to produce a basic model relatively quickly, or you can use the build options to experiment with different settings.

The LSVM node is similar to the SVM node, but it is linear and is better at handling a large number of records.

When the model has been built, you can:

- Browse the model nugget to display the relative importance of the input fields in building the model.
- Append a Table node to the model nugget to view the model output.

Example. A medical researcher has obtained a dataset containing characteristics of a number of human cell samples extracted from patients who were believed to be at risk of developing cancer. Analysis of the original data showed that many of the characteristics differed significantly between benign and malignant samples. The researcher wants to develop an LSVM model that can use the values of similar cell characteristics in samples from other patients to give an early indication of whether their samples might be benign or malignant.

- [LSVM Node Model Options](#)
- [LSVM Build Options](#)

Related information

- [About SVM](#)
- [How SVM Works](#)

- [Tuning an SVM Model](#)
-

LSVM Node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Calculate predictor importance. For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may take longer to calculate for some models, particularly when working with large datasets, and is off by default for some models as a result. Predictor importance is not available for decision list models. See [Predictor Importance](#) for more information.

Related information

- [About SVM](#)
 - [How SVM Works](#)
 - [Tuning an SVM Model](#)
-

LSVM Build Options

Model Settings

Include intercept. Including the intercept (the constant term in the model) can increase the overall accuracy of the solution. If you can assume the data passes through the origin, you can exclude the intercept.

Sorting order for categorical target. Specifies the sorting order for the categorical target. This setting is ignored for continuous targets.

Regression precision (epsilon). Used only if the measurement level of the target field is *Continuous*. Causes errors to be accepted provided that they are less than the value specified here. Increasing the value may result in faster modeling, but at the expense of accuracy.

Exclude records with any missing values. When set to True, a record is excluded if any single value is missing.

Penalty Settings

Penalty function. Specifies the type of penalty function that is used to reduce the likelihood of overfitting. The options are either L1 or L2.

L1 and L2 reduce the chance of overfitting by adding a penalty on the coefficients. The difference between them is that when there is a large number of features, L1 uses feature selection by setting some coefficients to 0 during model building. L2 does not have this ability so should not be used when you have a large number of features.

Penalty parameter (lambda). Specifies the penalty (regularization) parameter. This setting is enabled if the Penalty function is set.

Related information

- [About SVM](#)
 - [How SVM Works](#)
 - [Tuning an SVM Model](#)
-

LSVM Model Nugget (interactive output)

After running an LSVM model, the following output is available.

Model Information

The Model Information view provides key information about the model. The table identifies some high-level model settings, such as:

- The name of the target specified on the Fields tab
- The model building method specified on the [Model Selection](#) settings

- The number of predictors input
- The number of predictors in the final model
- The regularization type (L1 or L2)
- The penalty parameter (lambda). This is the regularization parameter.
- The regression precision (epsilon). Errors are accepted if they are less than this value. A higher value may result in faster modeling, but at the expense of accuracy. It is only used only if the measurement level of the target field is *Continuous*.
- The classification accuracy percentage. This is only applicable for Classification.
- The average squared error. This is only applicable for Regression.

Records Summary

The Records Summary view provides information about the number and percentage of records (cases) included and excluded from the model.

Predictor Importance

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

Predicted by Observed

This displays a binned scatterplot of the predicted values on the vertical axis by the observed values on the horizontal axis. Ideally, the points should lie on a 45-degree line; this view can tell you whether any records are predicted particularly badly by the model.

Note: Predictor importance may take longer to calculate for LSVM and SVM than for other types of models. Selecting this option may slow performance, particularly with large datasets.

Confusion Matrix

The confusion matrix, sometimes referred to as the summary table, shows the number of cases correctly and incorrectly assigned to each of the groups based on the LSVM analysis.

- [LSVM Model Settings](#)

Related information

- [Working with output](#)
- [LSVM Node](#)
- [LSVM Model Settings](#)

LSVM Model Settings

On the Settings tab for an SVLM model nugget, you specify options for raw propensity and for SQL generation during model scoring. This tab is available only after the model nugget is added to a stream.

Calculate raw propensity scores For models with flag targets only, you can request raw propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to standard prediction and confidence values. Adjusted propensity scores are not available.

Generate SQL for this model When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL is generated.

- Default: Score using Server Scoring Adapter (if installed) otherwise in process. If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS® Modeler.
- Score outside of the Database. If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

Nearest Neighbor Models

- [KNN node](#)
 - [KNN Model Nugget](#)
-

KNN node

Nearest Neighbor Analysis is a method for classifying cases based on their similarity to other cases. In machine learning, it was developed as a way to recognize patterns of data without requiring an exact match to any stored patterns, or cases. Similar cases are near each other and dissimilar cases are distant from each other. Thus, the distance between two cases is a measure of their dissimilarity.

Cases that are near each other are said to be “neighbors.” When a new case (holdout) is presented, its distance from each of the cases in the model is computed. The classifications of the most similar cases – the nearest neighbors – are tallied and the new case is placed into the category that contains the greatest number of nearest neighbors.

You can specify the number of nearest neighbors to examine; this value is called k . The pictures show how a new case would be classified using two different values of k . When $k = 5$, the new case is placed in category 1 because a majority of the nearest neighbors belong to category 1. However, when $k = 9$, the new case is placed in category 0 because a majority of the nearest neighbors belong to category 0.

Nearest neighbor analysis can also be used to compute values for a continuous target. In this situation, the average or median target value of the nearest neighbors is used to obtain the predicted value for the new case.

- [KNN Node Objectives Options](#)
 - [KNN Node Settings](#)
-

KNN Node Objectives Options

The Objectives tab is where you can choose either to build a model that predicts the value of a target field in your input data based on the values of its nearest neighbors, or to simply find which are the nearest neighbors for a particular case of interest.

What type of analysis do you want to perform?

Predict a target field. Choose this option if you want to predict the value of a target field based on the values of its nearest neighbors.

Only identify the nearest neighbors. Choose this option if you only want to see which are the nearest neighbors for a particular input field.

If you choose to identify only the nearest neighbors, the remaining options on this tab relating to accuracy and speed are disabled as they are relevant only for predicting targets.

What is your objective?

When predicting a target field, this group of options lets you decide whether speed, accuracy, or a blend of both, are the most important factors when predicting a target field. Alternatively you can choose to customize settings yourself.

If you choose the Balance, Speed, or Accuracy option, the algorithm preselects the most appropriate combination of settings for that option. Advanced users may wish to override these selections; this can be done on the various panels of the Settings tab.

Balance speed and accuracy. Selects the best number of neighbors within a small range.

Speed. Finds a fixed number of neighbors.

Accuracy. Selects the best number of neighbors within a larger range, and uses predictor importance when calculating distances.

Custom analysis. Choose this option to fine-tune the algorithm on the Settings tab.

Note: The size of the resulting KNN model, unlike most other models, increases linearly with the quantity of training data. If, when trying to build a KNN model, you see an error reporting an “out of memory” error, try increasing the maximum system memory used by IBM® SPSS® Modeler. To do so, choose

Tools > Options > System Options

and enter the new size in the Maximum memory field. Changes made in the System Options dialog do not take effect until you restart IBM SPSS Modeler.

Related information

- [KNN node](#)

KNN Node Settings

The Settings tab is where you specify the options that are specific to Nearest Neighbor Analysis. The sidebar on the left of the screen lists the panels that you use to specify the options.

- [Model](#)
- [Neighbors](#)
- [Feature Selection](#)
- [Cross-Validation](#)
- [Analyze](#)

Related information

- [KNN node](#)
- [Model](#)
- [Neighbors](#)
- [Feature Selection](#)
- [Cross-Validation](#)
- [Analyze](#)

Model

The Model panel provides options that control how the model is to be built, for example, whether to use partitioning or split models, whether to transform numeric input fields so that they all fall within the same range, and how to manage cases of interest. You can also choose a custom name for the model.

Note: The Use partitioned data and Use case labels can not use the same field.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Create split models. Builds a separate model for each possible value of input fields that are specified as split fields. See [Building Split Models](#) for more information.

To select fields manually... By default, the node uses the partition and split field settings (if any) from the Type node, but you can override those settings here. To activate the Partition and Splits fields, select the Fields tab and choose Use Custom Settings, then return here.

- Partition. This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to generate the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)
- Splits. For split models, select the split field or fields. This is similar to setting the field role to *Split* in a Type node. You can designate only fields of type Flag, Nominal or Ordinal as split fields. Fields chosen as split fields cannot be used as target, input, partition, frequency or weight fields. See the topic [Building Split Models](#) for more information.

Normalize range inputs. Check this box to normalize the values for continuous input fields. Normalized features have the same range of values, which can improve the performance of the estimation algorithm. Adjusted normalization, $[2*(x-\min)/(max-\min)]-1$, is used. Adjusted normalized values fall between -1 and 1.

Use case labels. Check this box to enable the drop-down list, from where you can choose a field whose values will be used as labels to identify the cases of interest in the predictor space chart, peers chart, and quadrant map in the model viewer. You can choose any field with a measurement level of *Nominal*, *Ordinal*, or *Flag* to use as the labeling field. If you do not choose a field here, records are displayed in the model viewer charts with nearest neighbors being identified by row number in the source data. If you will be manipulating the data at all after building the model, use case labels to avoid having to refer back to the source data each time to identify the cases in the display.

Identify focal record. Check this box to enable the drop-down list, which allows you to mark an input field of particular interest (for flag fields only). If you specify a field here, the points representing that field are initially selected in the model viewer when the model is built. Selecting a focal record here is optional; any point can temporarily become a focal record when selected manually in the model viewer.

Related information

- [KNN node](#)
 - [Neighbors](#)
 - [Feature Selection](#)
 - [Cross-Validation](#)
 - [Analyze](#)
-

Neighbors

The Neighbors panel has a set of options that control how the number of nearest neighbors is calculated.

Number of Nearest Neighbors (k). Specify the number of nearest neighbors for a particular case. Note that using a greater number of neighbors will not necessarily result in a more accurate model.

If the objective is to predict a target, you have two choices:

- **Specify fixed k.** Use this option if you want to specify a fixed number of nearest neighbors to find.
- **Automatically select k.** You can alternatively use the Minimum and Maximum fields to specify a range of values and allow the procedure to choose the "best" number of neighbors within that range. The method for determining the number of nearest neighbors depends upon whether feature selection is requested on the Feature Selection panel:
If feature selection is in effect, then feature selection is performed for each value of k in the requested range, and the k , and accompanying feature set, with the lowest error rate (or the lowest sum-of-squares error if the target is continuous) is selected.

If feature selection is not in effect, then V-fold cross-validation is used to select the "best" number of neighbors. See the Cross-validation panel for control over assignment of folds.

Distance Computation. This is the metric used to specify the distance metric used to measure the similarity of cases.

- **Euclidean metric.** The distance between two cases, x and y , is the square root of the sum, over all dimensions, of the squared differences between the values for the cases.
- **City Block metric.** The distance between two cases is the sum, over all dimensions, of the absolute differences between the values for the cases. Also called Manhattan distance.

Optionally, if the objective is to predict a target, you can choose to weight features by their normalized importance when computing distances. Feature importance for a predictor is calculated by the ratio of the error rate or sum-of-squares error of the model with the predictor removed from the model, to the error rate or sum-of-squares error for the full model. Normalized importance is calculated by reweighting the feature importance values so that they sum to 1.

Weight features by importance when computing distances. (Displayed only if the objective is to predict a target.) Check this box to cause predictor importance to be used when calculating the distances between neighbors. Predictor importance will then be displayed in the model nugget, and used in predictions (and so will affect scoring). See the topic [Predictor Importance](#) for more information.

Predictions for Range Target. (Displayed only if the objective is to predict a target.) If a continuous (numeric range) target is specified, this defines whether the predicted value is computed based upon the mean or the median value of the nearest neighbors.

Related information

- [KNN node](#)
 - [Model](#)
 - [Feature Selection](#)
 - [Cross-Validation](#)
 - [Analyze](#)
-

Feature Selection

This panel is activated only if the objective is to predict a target. It allows you to request and specify options for feature selection. By default, all features are considered for feature selection, but you can optionally select a subset of features to force into the model.

Perform feature selection. Check this box to enable the feature selection options.

- **Forced entry.** Click the field chooser button next to this box and choose one or more features to force into the model.

Stopping Criterion. At each step, the feature whose addition to the model results in the smallest error (computed as the error rate for a categorical target and sum of squares error for a continuous target) is considered for inclusion in the model set. Forward selection continues until the specified condition is met.

- **Stop when the specified number of features have been selected.** The algorithm adds a fixed number of features in addition to those forced into the model. Specify a positive integer. Decreasing values of the number to select creates a more parsimonious model, at the risk of missing important features. Increasing values of the number to select will capture all the important features, at the risk of eventually adding features that actually increase the model error.
- **Stop when the change in the absolute error ratio is less than or equal to the minimum.** The algorithm stops when the change in the absolute error ratio indicates that the model cannot be further improved by adding more features. Specify a positive number. Decreasing values of the minimum change will tend to include more features, at the risk of including features that do not add much value to the model. Increasing the value of the minimum change will tend to exclude more features, at the risk of losing features that are important to the model. The "optimal" value of the minimum change will depend upon your data and application. See the Feature Selection Error Log in the output to help you assess which features are most important. See the topic [Predictor selection error log \(Nearest Neighbor Analysis\)](#) for more information.

Related information

- [KNN node](#)
- [Model](#)
- [Neighbors](#)
- [Cross-Validation](#)
- [Analyze](#)

Cross-Validation

This panel is activated only if the objective is to predict a target. The options on this panel control whether to use cross-validation when calculating the nearest neighbors.

Cross-validation divides the sample into a number of subsamples, or **folds**. Nearest neighbor models are then generated, excluding the data from each subsample in turn. The first model is based on all of the cases except those in the first sample fold, the second model is based on all of the cases except those in the second sample fold, and so on. For each model, the error is estimated by applying the model to the subsample excluded in generating it. The "best" number of nearest neighbors is the one which produces the lowest error across folds.

Cross-Validation Folds. V-fold cross-validation is used to determine the "best" number of neighbors. It is not available in conjunction with feature selection for performance reasons.

- **Randomly assign cases to folds.** Specify the number of folds that should be used for cross-validation. The procedure randomly assigns cases to folds, numbered from 1 to V, the number of folds.
- **Set random seed.** When estimating the accuracy of a model based on a random percentage, this option allows you to duplicate the same results in another session. By specifying the starting value used by the random number generator, you can ensure the same records are assigned each time the node is executed. Enter the desired seed value. If this option is not selected, a different sample will be generated each time the node is executed.
- **Use field to assign cases.** Specify a numeric field that assigns each case in the active dataset to a fold. The field must be numeric and take values from 1 to V. If any values in this range are missing, and on any split fields if split models are in effect, this will cause an error.

Related information

- [KNN node](#)
- [Model](#)
- [Neighbors](#)
- [Feature Selection](#)
- [Analyze](#)

Analyze

The Analyze panel is activated only if the objective is to predict a target. You can use it to specify whether the model is to include additional variables to contain:

- probabilities for each possible target field value
- distances between a case and its nearest neighbors
- raw and adjusted propensity scores (for flag targets only)

Append all probabilities. If this option is checked, probabilities for each possible value of a nominal or flag target field are displayed for each record processed by the node. If this option is unchecked, only the predicted value and its probability are displayed for nominal or flag target fields.

Save distances between cases and k nearest neighbors. For each focal record, a separate variable is created for each of the focal record's k nearest neighbors (from the training sample) and the corresponding k nearest distances.

Propensity Scores

Propensity scores can be enabled in the modeling node, and on the Settings tab in the model nugget. This functionality is available only when the selected target is a flag field. See the topic [Propensity Scores](#) for more information.

Calculate raw propensity scores. Raw propensity scores are derived from the model based on the training data only. If the model predicts the true value (will respond), then the propensity is the same as P , where P is the probability of the prediction. If the model predicts the false value, then the propensity is calculated as $(1 - P)$.

- If you choose this option when building the model, propensity scores will be enabled in the model nugget by default. However, you can always choose to enable raw propensity scores in the model nugget whether or not you select them in the modeling node.
- When scoring the model, raw propensity scores will be added in a field with the letters RP appended to the standard prefix. For example, if the predictions are in a field named $\$R\text{-churn}$, the name of the propensity score field will be $\$RRP\text{-churn}$.

Calculate adjusted propensity scores. Raw propensities are based purely on estimates given by the model, which may be overfitted, leading to over-optimistic estimates of propensity. Adjusted propensities attempt to compensate by looking at how the model performs on the test or validation partitions and adjusting the propensities to give a better estimate accordingly.

- This setting requires that a valid partition field is present in the stream.
- Unlike raw confidence scores, adjusted propensity scores must be calculated when building the model; otherwise, they will not be available when scoring the model nugget.
- When scoring the model, adjusted propensity scores will be added in a field with the letters AP appended to the standard prefix. For example, if the predictions are in a field named $\$R\text{-churn}$, the name of the propensity score field will be $\$RAP\text{-churn}$. Adjusted propensity scores are not available for logistic regression models.
- When calculating the adjusted propensity scores, the test or validation partition used for the calculation must not have been balanced. To avoid this, be sure the Only balance training data option is selected in any upstream Balance nodes. In addition, if a complex sample has been taken upstream this will invalidate the adjusted propensity scores.
- Adjusted propensity scores are not available for "boosted" tree and rule set models. See the topic [Boosted C5.0 Models](#) for more information.

Related information

- [KNN node](#)
- [Model](#)
- [Neighbors](#)
- [Feature Selection](#)
- [Cross-Validation](#)

KNN Model Nugget

The KNN model creates a number of new fields, as shown in the following table. To see these fields and their values, add a Table node to the KNN model nugget and execute the Table node, or click the Preview button on the nugget.

Table 1. KNN model fields

New field name	Description
$\$KNN\text{-fieldname}$	Predicted value of target field.
$\$KNNP\text{-fieldname}$	Probability of predicted value.
$\$KNNP\text{-value}$	Probability of each possible value of a nominal or flag field. Included only if Append all probabilities is checked on the Settings tab of the model nugget.
$\$KNN\text{-neighbor-}n$	The name of the n th nearest neighbor to the focal record. Included only if Display Nearest on the Settings tab of the model nugget is set to a non-zero value.
$\$KNN\text{-distance-}n$	The relative distance from the focal record of the n th nearest neighbor to the focal record. Included only if Display Nearest on the Settings tab of the model nugget is set to a non-zero value.

- [Nearest Neighbor Model View](#)
- [KNN Model Settings](#)

Nearest Neighbor Model View

- [Model View \(Nearest Neighbor Analysis\)](#)
-

Model View (Nearest Neighbor Analysis)

The model view has a 2-panel window:

- The first panel displays an overview of the model called the main view.
- The second panel displays one of two types of views:
An auxiliary model view shows more information about the model, but is not focused on the model itself.

A linked view is a view that shows details about one feature of the model when the user drills down on part of the main view.

By default, the first panel shows the predictor space and the second panel shows the predictor importance chart. If the predictor importance chart is not available; that is, when Weight features by importance was not selected on the Neighbors panel of the Settings tab, the first available view in the View dropdown is shown.

When a view has no available information, it is omitted from the View dropdown.

- [Predictor Space \(Nearest Neighbor Analysis\)](#)
 - [Predictor Importance \(Nearest Neighbor Analysis\)](#)
 - [Nearest Neighbor Distances \(Nearest Neighbor Analysis\)](#)
 - [Peers \(Nearest Neighbor Analysis\)](#)
 - [Quadrant Map \(Nearest Neighbor Analysis\)](#)
 - [Predictor selection error log \(Nearest Neighbor Analysis\)](#)
 - [Classification Table \(Nearest Neighbor Analysis\)](#)
 - [Error Summary \(Nearest Neighbor Analysis\)](#)
-

Predictor Space (Nearest Neighbor Analysis)

The predictor space chart is an interactive graph of the predictor space (or a subspace, if there are more than 3 predictors). Each axis represents a predictor in the model, and the location of points in the chart show the values of these predictors for cases in the training and holdout partitions.

Keys. In addition to the predictor values, points in the plot convey other information.

- Shape indicates the partition to which a point belongs, either Training or Holdout.
- The color/shading of a point indicates the value of the target for that case; with distinct color values equal to the categories of a categorical target, and shades indicating the range of values of a continuous target. The indicated value for the training partition is the observed value; for the holdout partition, it is the predicted value. If no target is specified, this key is not shown.
- Heavier outlines indicate a case is focal. Focal records are shown linked to their k nearest neighbors.

Controls and Interactivity. A number of controls in the chart allow you explore the predictor space.

- You can choose which subset of predictors to show in the chart and change which predictors are represented on the dimensions.
 - “Focal records” are simply points selected in the Predictor Space chart. If you specified a focal record variable, the points representing the focal records will initially be selected. However, any point can temporarily become a focal record if you select it. The “usual” controls for point selection apply; clicking on a point selects that point and deselects all others; Control-clicking on a point adds it to the set of selected points. Linked views, such as the Peers Chart, will automatically update based upon the cases selected in the Predictor Space.
 - You can change the number of nearest neighbors (k) to display for focal records.
 - Hovering over a point in the chart displays a tooltip with the value of the case label, or case number if case labels are not defined, and the observed and predicted target values.
 - A “Reset” button allows you to return the Predictor Space to its original state.
 - [Changing the axes on the Predictor Space chart](#)
-

Changing the axes on the Predictor Space chart

You can control which features are displayed on the axes of the Predictor Space chart.

To change the axis settings:

1. Click the Edit Mode button (paintbrush icon) in the left-hand panel to select Edit mode for the Predictor Space.

2. Change the view (to anything) on the right-hand panel. The Show zones panel appears between the two main panels.
3. Click the Show zones check box.
4. Click any data point in the Predictor Space.
5. To replace an axis with a predictor of the same data type:
 - Drag the new predictor over the zone label (the one with the small X button) of the one you want to replace.
6. To replace an axis with a predictor of a different data type:
 - On the zone label of the predictor you want to replace, click the small X button. The predictor space changes to a two-dimensional view.
 - Drag the new predictor over the Add dimension zone label.
7. Click the Explore Mode button (arrowhead icon) in the left-hand panel to exit from Edit mode.

Predictor Importance (Nearest Neighbor Analysis)

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

Nearest Neighbor Distances (Nearest Neighbor Analysis)

This table displays the k nearest neighbors and distances for focal records only. It is available if a focal record identifier is specified on the modeling node, and only displays focal records identified by this variable.

Each row of:

- The Focal Record column contains the value of the case labeling variable for the focal record; if case labels are not defined, this column contains the case number of the focal record.
- The i th column under the Nearest Neighbors group contains the value of the case labeling variable for the i th nearest neighbor of the focal record; if case labels are not defined, this column contains the case number of the i th nearest neighbor of the focal record.
- The i th column under the Nearest Distances group contains the distance of the i th nearest neighbor to the focal record

Peers (Nearest Neighbor Analysis)

This chart displays the focal cases and their k nearest neighbors on each predictor and on the target. It is available if a focal case is selected in the Predictor Space.

The Peers chart is linked to the Predictor Space in two ways.

- Cases selected (focal) in the Predictor Space are displayed in the Peers chart, along with their k nearest neighbors.
- The value of k selected in the Predictor Space is used in the Peers chart.

Select Predictors. Enables you to select the predictors to display in the Peers chart.

Quadrant Map (Nearest Neighbor Analysis)

This chart displays the focal cases and their k nearest neighbors on a scatterplot (or dotplot, depending upon the measurement level of the target) with the target on the y -axis and a scale predictor on the x -axis, paneled by predictors. It is available if there is a target and if a focal case is selected in the Predictor Space.

- Reference lines are drawn for continuous variables, at the variable means in the training partition.

Select Predictors. Enables you to select the predictors to display in the Quadrant Map.

Predictor selection error log (Nearest Neighbor Analysis)

Points on the chart display the error (either the error rate or sum-of-squares error, depending upon the measurement level of the target) on the y -axis for the model with the predictor listed on the x -axis (plus all features to the left on the x -axis). This chart is available if there is a target and

feature selection is in effect.

Classification Table (Nearest Neighbor Analysis)

This table displays the cross-classification of observed versus predicted values of the target, by partition. It is available if there is a target and it is categorical (flag, nominal, or ordinal).

- The (Missing) row in the Holdout partition contains holdout cases with missing values on the target. These cases contribute to the Holdout Sample: Overall Percent values but not to the Percent Correct values.

Error Summary (Nearest Neighbor Analysis)

This table is available if there is a target variable. It displays the error associated with the model; sum-of-squares for a continuous target and the error rate (100% – overall percent correct) for a categorical target.

KNN Model Settings

The Settings tab enables you to specify extra fields to be displayed when viewing the results (for example by executing a Table node attached to the nugget). You can see the effect of each of these options by selecting them and clicking the Preview button--scroll to the right of the Preview output to see the extra fields.

Append all probabilities (valid only for categorical targets). If this option is checked, probabilities for each possible value of a nominal or flag target field are displayed for each record processed by the node. If this option is unchecked, only the predicted value and its probability are displayed for nominal or flag target fields.

The default setting of this check box is determined by the corresponding check box on the modeling node.

Calculate raw propensity scores. For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

Calculate adjusted propensity scores. Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

Display nearest. If you set this value to n , where n is a non-zero positive integer, the n nearest neighbors to the focal record are included in the model, together with their relative distances from the focal record.

Glossary

A

AICC

A measure for selecting and comparing mixed models based on the -2 (Restricted) log likelihood. Smaller values indicate better models. The AICC "corrects" the AIC for small sample sizes. As the sample size increases, the AICC converges to the AIC.

B

Bayesian Information Criterion (BIC)

A measure for selecting and comparing models based on the -2 log likelihood. Smaller values indicate better models. The BIC also "penalizes" overparameterized models (complex models with a large number of inputs, for example), but more strictly than the AIC.

Box's M test

A test for the equality of the group covariance matrices. For sufficiently large samples, a nonsignificant p value means there is insufficient evidence that the matrices differ. The test is sensitive to departures from multivariate normality.

C

Cases

Codes for actual group, predicted group, posterior probabilities, and discriminant scores are displayed for each case.

Classification Results

The number of cases correctly and incorrectly assigned to each of the groups based on the discriminant analysis. Sometimes called the "Confusion Matrix."

Combined-Groups Plots

Creates an all-groups scatterplot of the first two discriminant function values. If there is only one function, a histogram is displayed instead.

Covariance

An unstandardized measure of association between two variables, equal to the cross-product deviation divided by N-1.

F

Fisher's

Displays Fisher's classification function coefficients that can be used directly for classification. A separate set of classification function coefficients is obtained for each group, and a case is assigned to the group for which it has the largest discriminant score (classification function value).

H

Hazard Plot

Displays the cumulative hazard function on a linear scale.

K

Kurtosis

A measure of the extent to which there are outliers. For a normal distribution, the value of the kurtosis statistic is zero. Positive kurtosis indicates that the data exhibit more extreme outliers than a normal distribution. Negative kurtosis indicates that the data exhibit less extreme outliers than a normal distribution.

L

Leave-one-out Classification

Each case in the analysis is classified by the functions derived from all cases other than that case. It is also known as the "U-method."

M

MAE

Mean absolute error. Measures how much the series varies from its model-predicted level. MAE is reported in the original series units.

Mahalanobis Distance

A measure of how much a case's values on the independent variables differ from the average of all cases. A large Mahalanobis distance identifies a case as having extreme values on one or more of the independent variables.

MAPE

Mean Absolute Percentage Error. A measure of how much a dependent series varies from its model-predicted level. It is independent of the units used and can therefore be used to compare series with different units.

MaxAE

Maximum Absolute Error. The largest forecasted error, expressed in the same units as the dependent series. Like MaxAPE, it is useful for imagining the worst-case scenario for your forecasts. Maximum absolute error and maximum absolute percentage error may occur at different series points--for example, when the absolute error for a large series value is slightly larger than the absolute error for a small series value. In

that case, the maximum absolute error will occur at the larger series value and the maximum absolute percentage error will occur at the smaller series value.

MaxAPE

Maximum Absolute Percentage Error. The largest forecasted error, expressed as a percentage. This measure is useful for imagining a worst-case scenario for your forecasts.

Maximizing the Smallest F Ratio Method of Entry

A method of variable selection in stepwise analysis based on maximizing an F ratio computed from the Mahalanobis distance between groups.

Maximum

The largest value of a numeric variable.

Mean

A measure of central tendency. The arithmetic average, the sum divided by the number of cases.

Means

Displays total and group means, as well as standard deviations for the independent variables.

Median

The value above and below which half of the cases fall, the 50th percentile. If there is an even number of cases, the median is the average of the two middle cases when they are sorted in ascending or descending order. The median is a measure of central tendency not sensitive to outlying values (unlike the mean, which can be affected by a few extremely high or low values).

Minimize Wilks' Lambda

A variable selection method for stepwise discriminant analysis that chooses variables for entry into the equation on the basis of how much they lower Wilks' lambda. At each step, the variable that minimizes the overall Wilks' lambda is entered.

Minimum

The smallest value of a numeric variable.

Mode

The most frequently occurring value. If several values share the greatest frequency of occurrence, each of them is a mode.

N

Normalized BIC

Normalized Bayesian Information Criterion. A general measure of the overall fit of a model that attempts to account for model complexity. It is a score based upon the mean square error and includes a penalty for the number of parameters in the model and the length of the series. The penalty removes the advantage of models with more parameters, making the statistic easy to compare across different models for the same series.

O

One Minus Survival

Plots one-minus the survival function on a linear scale.

R

Range

The difference between the largest and smallest values of a numeric variable, the maximum minus the minimum.

Rao's V (Discriminant Analysis)

A measure of the differences between group means. Also called the Lawley-Hotelling trace. At each step, the variable that maximizes the increase in Rao's V is entered. After selecting this option, enter the minimum value a variable must have to enter the analysis.

RMSE

Root Mean Square Error. The square root of mean square error. A measure of how much a dependent series varies from its model-predicted level, expressed in the same units as the dependent series.

R-Squared

Goodness-of-fit measure of a linear model, sometimes called the coefficient of determination. It is the proportion of variation in the dependent variable explained by the regression model. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well.

S

Separate-Groups

Separate-groups covariance matrices are used for classification. Because classification is based on the discriminant functions (not based on the original variables), this option is not always equivalent to quadratic discrimination.

Separate-Groups Covariance

Displays separate covariance matrices for each group.

Separate-Groups Plots

Creates separate-group scatterplots of the first two discriminant function values. If there is only one function, histograms are displayed instead.

Sequential Bonferroni

This is a sequentially step-down rejective Bonferroni procedure that is much less conservative in terms of rejecting individual hypotheses but maintains the same overall significance level.

Sequential Sidak

This is a sequentially step-down rejective Sidak procedure that is much less conservative in terms of rejecting individual hypotheses but maintains the same overall significance level.

Skewness

A measure of the asymmetry of a distribution. The normal distribution is symmetric and has a skewness value of 0. A distribution with a significant positive skewness has a long right tail. A distribution with a significant negative skewness has a long left tail. As a guideline, a skewness value more than twice its standard error is taken to indicate a departure from symmetry.

standard deviation

A measure of dispersion around the mean, equal to the square root of the variance. The standard deviation is measured in the same units as the original variable.

Standard Deviation

A measure of dispersion around the mean. In a normal distribution, 68% of cases fall within one standard deviation of the mean and 95% of cases fall within two standard deviations. For example, if the mean age is 45, with a standard deviation of 10, 95% of the cases would be between 25 and 65 in a normal distribution.

Standard Error

A measure of how much the value of a test statistic varies from sample to sample. It is the standard deviation of the sampling distribution for a statistic. For example, the standard error of the mean is the standard deviation of the sample means.

Standard Error of Kurtosis

The ratio of kurtosis to its standard error can be used as a test of normality (that is, you can reject normality if the ratio is less than -2 or greater than +2). A large positive value for kurtosis indicates that the tails of the distribution are longer than those of a normal distribution; a negative value for kurtosis indicates shorter tails (becoming like those of a box-shaped uniform distribution).

Standard Error of Mean

A measure of how much the value of the mean may vary from sample to sample taken from the same distribution. It can be used to roughly compare the observed mean to a hypothesized value (that is, you can conclude the two values are different if the ratio of the difference to the standard error is less than -2 or greater than +2).

Standard Error of Skewness

The ratio of skewness to its standard error can be used as a test of normality (that is, you can reject normality if the ratio is less than -2 or greater than +2). A large positive value for skewness indicates a long right tail; an extreme negative value indicates a long left tail.

Stationary R-squared

A measure that compares the stationary part of the model to a simple mean model. This measure is preferable to ordinary R-squared when there is a trend or seasonal pattern. Stationary R-squared can be negative with a range of negative infinity to 1. Negative values mean that the model under consideration is worse than the baseline model. Positive values mean that the model under consideration is better than the baseline model.

Sum

The sum or total of the values, across all cases with nonmissing values.

Survival Plot

Displays the cumulative survival function on a linear scale.

T

Territorial Map

A plot of the boundaries used to classify cases into groups based on function values. The numbers correspond to groups into which cases are classified. The mean for each group is indicated by an asterisk within its boundaries. The map is not displayed if there is only one discriminant function.

Total Covariance

Displays a covariance matrix from all cases as if they were from a single sample.

U

Unexplained Variance

At each step, the variable that minimizes the sum of the unexplained variation between groups is entered.

Unique

Evaluates all effects simultaneously, adjusting each effect for all other effects of any type.

Univariate ANOVAs

Performs a one-way analysis-of-variance test for equality of group means for each independent variable.

Unstandardized

Displays the unstandardized discriminant function coefficients.

Use F Value

A variable is entered into the model if its F value is greater than the Entry value and is removed if the F value is less than the Removal value. Entry must be greater than Removal, and both values must be positive. To enter more variables into the model, lower the Entry value. To remove more variables from the model, increase the Removal value.

Use Probability of F

A variable is entered into the model if the significance level of its F value is less than the Entry value and is removed if the significance level is greater than the Removal value. Entry must be less than Removal, and both values must be positive. To enter more variables into the model, increase the Entry value. To remove more variables from the model, lower the Removal value.

V

Valid

Valid cases having neither the system-missing value, nor a value defined as user-missing.

Variance

A measure of dispersion around the mean, equal to the sum of squared deviations from the mean divided by one less than the number of cases. The variance is measured in units that are the square of those of the variable itself.

W

Within-Groups

The pooled within-groups covariance matrix is used to classify cases.

Within-Groups Correlation

Displays a pooled within-groups correlation matrix that is obtained by averaging the separate covariance matrices for all groups before computing the correlations.

Within-Groups Covariance

Displays a pooled within-groups covariance matrix, which may differ from the total covariance matrix. The matrix is obtained by averaging the separate covariance matrices for all groups.

Python nodes

SPSS® Modeler offers nodes for using Python native algorithms. The Python tab on the [Nodes Palette](#) contains the following nodes you can use to run Python algorithms. These nodes are supported on Windows 64, Linux64, and Mac.

	The Synthetic Minority Over-sampling Technique (SMOTE) node provides an over-sampling algorithm to deal with imbalanced data sets. It provides an advanced method for balancing data. The SMOTE process node in SPSS Modeler is implemented in Python and requires the imbalanced-learn® Python library.
	XGBoost Linear® is an advanced implementation of a gradient boosting algorithm with a linear model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. The XGBoost Linear node in SPSS Modeler is implemented in Python.
	XGBoost Tree® is an advanced implementation of a gradient boosting algorithm with a tree model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost Tree is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost Tree node in SPSS Modeler exposes the core features and commonly used parameters. The node is implemented in Python.
	t-Distributed Stochastic Neighbor Embedding (t-SNE) is a tool for visualizing high-dimensional data. It converts affinities of data points to probabilities. This t-SNE node in SPSS Modeler is implemented in Python and requires the scikit-learn® Python library.
	A Gaussian Mixture® model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians. The Gaussian Mixture node in SPSS Modeler exposes the core features and commonly used parameters of the Gaussian Mixture library. The node is implemented in Python.
	Kernel Density Estimation (KDE)® uses the Ball Tree or KD Tree algorithms for efficient queries, and combines concepts from unsupervised learning, feature engineering, and data modeling. Neighbor-based approaches such as KDE are some of the most popular and useful density estimation techniques. The KDE Modeling and KDE Simulation nodes in SPSS Modeler expose the core features and commonly used parameters of the KDE library. The nodes are implemented in Python.
	The Random Forest node uses an advanced implementation of a bagging algorithm with a tree model as the base model. This Random Forest modeling node in SPSS Modeler is implemented in Python and requires the scikit-learn® Python library.
	Hierarchical Density-Based Spatial Clustering (HDBSCAN)® uses unsupervised learning to find clusters, or dense regions, of a data set. The HDBSCAN node in SPSS Modeler exposes the core features and commonly used parameters of the HDBSCAN library. The node is implemented in Python, and you can use it to cluster your dataset into distinct groups when you don't know what those groups are at first.
	The One-Class SVM node uses an unsupervised learning algorithm. The node can be used for novelty detection. It will detect the soft boundary of a given set of samples, to then classify new points as belonging to that set or not. This One-Class SVM modeling node in SPSS Modeler is implemented in Python and requires the scikit-learn® Python library.

- [SMOTE node](#)
- [XGBoost Linear node](#)
- [XGBoost Tree node](#)
- [t-SNE node](#)
- [Gaussian Mixture node](#)
- [KDE nodes](#)
- [Random Forest node](#)
- [HDBSCAN node](#)
- [One-Class SVM node](#)

SMOTE node

The Synthetic Minority Over-sampling Technique (SMOTE) node provides an over-sampling algorithm to deal with imbalanced data sets. It provides an advanced method for balancing data. The SMOTE process node in SPSS Modeler is implemented in Python and requires the imbalanced-learn®

Python library. For details about the imbalanced-learn library, see <https://imbalanced-learn.org/stable/>¹.

The Python tab on the Nodes Palette contains the SMOTE node and other Python nodes.

¹Lemaître, Nogueira, Aridas. "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning." *Journal of Machine Learning Research*, vol. 18, no. 17, 2017, pp. 1-5. (<http://jmlr.org/papers/v18/16-365.html>)

- [SMOTE node Settings](#)
- [SMOTE node Settings](#)

SMOTE node Settings

Define the following settings on the SMOTE node's Settings tab.

Target Setting

Target Field. Select the target field. All flag, nominal, ordinal, and discrete measurement types are supported. If the Use partitioned data option is selected in the Partition section, only training data will be over-sampled.

Over Sample Ratio

Select Auto to automatically select an over-sample ratio, or select Set Ratio (minority over majority) to set a custom ratio value. The ratio is the number of samples in the minority class over the number of samples in the majority class. The value must be greater than 0 and less than or equal to 1.

Random Seed

Set random seed. Select this option and click Generate to generate the seed used by the random number generator.

Methods

Algorithm Kind. Select the type of SMOTE algorithm you wish to use.

Samples Rules

K Neighbours. Specify the number of the nearest neighbors to use for constructing synthetic samples

M Neighbours. Specify the number of nearest neighbors to use for determining if a minority sample is in danger. This will only be used if the Borderline1 or Borderline2 SMOTE algorithm type is selected.

Partition

Use partitioned data. Select this option if you only want training data to be over-sampled.

The SMOTE node requires the imbalanced-learn® Python library. The following table shows the relationship between the settings in the SPSS® Modeler SMOTE node dialog and the Python algorithm.

Table 1. Node properties mapped to Python library parameters

SPSS Modeler setting	Script name (property name)	Python API parameter name
Over sample ratio (number input control)	sample_ratio_value	ratio
Random seed	random_seed	random_state
K_Neighbours	k_neighbours	k
M_Neighbours	m_neighbours	m
Algorithm kind	algorithm_kind	kind

XGBoost Linear node

XGBoost Linear® is an advanced implementation of a gradient boosting algorithm with a linear model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. The XGBoost Linear node in SPSS® Modeler is implemented in Python.

For more information about boosting algorithms, see the XGBoost Tutorials available at <http://xgboost.readthedocs.io/en/latest/tutorials/index.html>.¹

Note that the XGBoost cross-validation function is not supported in SPSS Modeler. You can use the SPSS Modeler Partition node for this functionality. Also note that XGBoost in SPSS Modeler performs one-hot encoding automatically for categorical variables.

¹ "XGBoost Tutorials." *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

- [XGBoost Linear node Fields](#)
 - [XGBoost Linear node Build Options](#)
 - [XGBoost Linear node Model Options](#)
-

XGBoost Linear node Fields

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign a target and predictors, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the Target and Predictors role fields on the right of the screen. The icons indicate the valid measurement levels for each role field. To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Target. Select a field to use as the target for the prediction.

Predictors. Select one or more fields as inputs for the prediction.

XGBoost Linear node Build Options

Use the Build Options tab to specify build options for the XGBoost Linear node, including basic options such as linear boost parameters and model building, and learning task options for objectives. For additional information about these options, see the following online resources:

- [XGBoost Parameter Reference](#)¹
- [XGBoost Python API](#)²
- [XGBoost Home Page](#)³

Basic

Hyper-Parameter Optimization (Based on Rbfopt). Select this option to enable Hyper-Parameter Optimization based on Rbfopt, which automatically discovers the optimal combination of parameters so that the model will achieve the expected or lower error rate on the samples. For details about Rbfopt, see http://rbfopt.readthedocs.io/en/latest/rbfopt_settings.html.

Alpha. L1 regularization term on weights. Increasing this value will make model more conservative.

Lambda. L2 regularization term on weights. Increasing this value will make the model more conservative.

Lambda bias. L2 regularization term on bias. (There is no L1 regularization term on bias because it is not important.)

Number boost round. The number of boosting iterations.

Learning Task

Objective. Select from the following learning task objective types: reg:linear, reg:logistic, reg:gamma, reg:tweedie, count:poisson, rank:pairwise, binary:logistic, or multi.

Random Seed. You can click Generate to generate the seed used by the random number generator.

The following table shows the relationship between the settings in the SPSS® Modeler XGBoost Linear node dialog and the Python XGBoost library parameters.

Table 1. Node properties mapped to Python library parameters

SPSS Modeler setting	Script name (property name)	XGBoost parameter
Target	TargetField	

SPSS Modeler setting	Script name (property name)	XGBoost parameter
Predictors	InputFields	
Lambda	lambda	lambda
Alpha	alpha	alpha
Lambda bias	lambdaBias	lambda_bias
Num boost round	numBoostRound	num_boost_round
Objective	objectiveType	objective
Random Seed	random_seed	seed

¹ "XGBoost Parameters" *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

² "Plotting API" *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

³ "Scalable and Flexible Gradient Boosting." Web. © 2015-2016 DMLC.

XGBoost Linear node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

XGBoost Tree node

XGBoost Tree[©] is an advanced implementation of a gradient boosting algorithm with a tree model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost Tree is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost Tree node in SPSS® Modeler exposes the core features and commonly used parameters. The node is implemented in Python.

For more information about boosting algorithms, see the XGBoost Tutorials available at

<http://xgboost.readthedocs.io/en/latest/tutorials/index.html>.¹

Note that the XGBoost cross-validation function is not supported in SPSS Modeler. You can use the SPSS Modeler Partition node for this functionality. Also note that XGBoost in SPSS Modeler performs one-hot encoding automatically for categorical variables.

¹ "XGBoost Tutorials." *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

- [XGBoost Tree node Fields](#)
- [XGBoost Tree node Build Options](#)
- [XGBoost Tree node Model Options](#)

XGBoost Tree node Fields

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign a target and predictors, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the Target and Predictors role fields on the right of the screen. The icons indicate the valid measurement levels for each role field. To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Target. Select a field to use as the target for the prediction.

Predictors. Select one or more fields as inputs for the prediction.

XGBoost Tree node Build Options

Use the Build Options tab to specify build options for the XGBoost Tree node, including basic options for model building and tree growth, learning task options for objectives, and advanced options for control overfitting and handling of imbalanced datasets. For additional information about these options, see the following online resources:

- [XGBoost Parameter Reference](#)¹
- [XGBoost Python API](#)²
- [XGBoost Home Page](#)³

Basic

Hyper-Parameter Optimization (Based on Rbfopt). Select this option to enable Hyper-Parameter Optimization based on Rbfopt, which automatically discovers the optimal combination of parameters so that the model will achieve the expected or lower error rate on the samples. For details about Rbfopt, see http://rbfopt.readthedocs.io/en/latest/rbfopt_settings.html.

Tree method. Select the XGBoost tree construction algorithm to use.

Num boost round. Specify the number of boosting iterations.

Max depth. Specify the maximum depth for trees. Increasing this value will make the model more complex and likely to be overfitting.

Min child weight. Specify the minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than this Min child weight, then the building process will stop further partitioning. In linear regression mode, this simply corresponds to minimum number of instances needed in each node. The larger the weight, the more conservative the algorithm will be.

Max delta step. Specify the maximum delta step to allow for each tree's weight estimation. If set to 0, there is no constraint. If set to a positive value, it can help the update step be more conservative. Usually this parameter is not needed, but it may help in logistic regression when a class is extremely imbalanced.

Learning Task

Objective. Select from the following learning task objective types: reg:linear, reg:logistic, reg:gamma, reg:tweedie, count:poisson, rank:pairwise, binary:logistic, or multi.

Early stopping. Select this option if you want to use the early stopping function. For the stopping rounds, validation errors must decrease at least every early stopping round(s) to continue training. The Evaluation data ratio is the ratio of input data used for validation errors.

Random Seed. You can click Generate to generate the seed used by the random number generator.

Advanced

Sub sample. Sub sample is the ratio of the training instance. For example, if you set this to 0.5, XGBoost will randomly collect half the data instances to grow trees and this will prevent overfitting.

Eta. The step size shrinkage used during the update step to prevent overfitting. After each boosting step, the weights of new features can be obtained directly. Eta also shrinks the feature weights to make the boosting process more conservative.

Gamma. The minimum loss reduction required to make a further partition on a leaf node of the tree. The larger the gamma setting, the more conservative the algorithm will be.

Colsample by tree. Sub sample ratio of columns when constructing each tree.

Colsample by level. Sub sample ratio of columns for each split, in each level.

Lambda. L2 regularization term on weights. Increasing this value will make the model more conservative.

Alpha. L1 regularization term on weights. Increasing this value will make model more conservative.

Scale pos weight. Control the balance of positive and negative weights. This is useful for unbalanced classes.

The following table shows the relationship between the settings in the SPSS® Modeler XGBoost Tree node dialog and the Python XGBoost library parameters.

Table 1. Node properties mapped to Python library parameters

SPSS Modeler setting	Script name (property name)	XGBoost parameter
Target	TargetField	
Predictors	InputFields	
Tree method	treeMethod	tree_method
Num boost round	numBoostRound	num_boost_round
Max depth	maxDepth	max_depth
Min child weight	minChildWeight	min_child_weight

SPSS Modeler setting	Script name (property name)	XGBoost parameter
Max delta step	maxDeltaStep	max_delta_step
Objective	objectiveType	objective
Early stopping	earlyStopping	early_stopping_rounds
stopping rounds	stoppingRounds	
Evaluation data ratio	evaluationDataRatio	
Random Seed	random_seed	seed
Sub sample	sampleSize	subsample
Eta	eta	eta
Gamma	gamma	gamma
Colsample by tree	colSampleRatio	colsample_bytree
Colsample by level	colSampleLevel	colsample_bylevel
Lambda	lambda	lambda
Alpha	alpha	alpha
Scale pos weight	scalePosWeight	scale_pos_weight

¹ "XGBoost Parameters" *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

² "Plotting API" *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

³ "Scalable and Flexible Gradient Boosting." Web. © 2015-2016 DMLC.

XGBoost Tree node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

t-SNE node

t-Distributed Stochastic Neighbor Embedding (t-SNE)© is a tool for visualizing high-dimensional data. It converts affinities of data points to probabilities. The affinities in the original space are represented by Gaussian joint probabilities and the affinities in the embedded space are represented by Student's t-distributions. This allows t-SNE to be particularly sensitive to local structure and has a few other advantages over existing techniques:¹

- Revealing the structure at many scales on a single map
- Revealing data that lie in multiple, different, manifolds, or clusters
- Reducing the tendency to crowd points together at the center

The t-SNE node in SPSS® Modeler is implemented in Python and requires the scikit-learn© Python library. For details about t-SNE and the scikit-learn library, see:

- <https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html#sklearn.manifold.TSNE>
- <https://scikit-learn.org/stable/modules/manifold.html#t-sne>

The Python tab on the Nodes Palette contains this node and other Python nodes. The t-SNE node is also available on the Graphs tab.

¹ References:

van der Maaten, L.J.P.; Hinton, G. ["Visualizing High-Dimensional Data using t-SNE."](#) Journal of Machine Learning Research. 9:2579-2605, 2008.

van der Maaten, L.J.P. ["t-Distributed Stochastic Neighbor Embedding."](#)

van der Maaten, L.J.P. ["Accelerating t-SNE using Tree-Based Algorithms."](#) Journal of Machine Learning Research. 15(Oct):3221-3245, 2014.

- [t-SNE node Expert options](#)
- [t-SNE node Output options](#)
- [Accessing and plotting t-SNE data](#)
- [t-SNE model nuggets](#)
- [t-SNE node Expert options](#)
- [t-SNE node Output options](#)
- [t-SNE model nuggets](#)

t-SNE node Expert options

Choose Simple mode or Expert mode depending on which options you want to set for the t-SNE node.

Visualization type. Select 2D or 3D to specify whether to draw the graph as two-dimensional or three-dimensional.

Method. Select Barnes Hut or Exact. By default, the gradient calculation algorithm uses Barnes-Hut approximation which runs much faster than the Exact method. Barnes-Hut approximation allows the t-SNE technique to be applied to large, real-world datasets. The Exact algorithm will do a better job of avoiding nearest-neighbor errors.

Init. Select Random or PCA for the initialization of embedding.

Target Field. Select the target field to show as a colormap on the output graph. The graph will use one color if no target field is specified here.

Optimization

Perplexity. The perplexity is related to the number of nearest neighbors that are used in other manifold learning algorithms. Larger datasets usually require a larger perplexity. Consider selecting a value between 5 and 50. Default is 30, and the range is 2 - 9999999.

Early exaggeration. This setting controls how tight natural clusters in the original space will be in the embedded space, and how much space will be between them. Default is 12, and the range is 2 - 9999999.

Learning rate. If the learning rate is too high, the data might look like a "ball" with any point approximately equidistant from its nearest neighbors. If the learning rate is too low, most points may look compressed in a dense cloud with few outliers. If the cost function gets stuck in a bad local minimum, increasing the learning rate may help. Default is 200, and the range is 0 - 9999999.

Max iterations. The maximum number of iterations for the optimization. Default is 1000, and the range is 250 - 9999999.

Angular size. The angular size of a distant node as measured from a point. Enter a value between 0 and 1. Default is 0.5.

Random seed

Set random seed. Select this option and click Generate to generate the seed used by the random number generator.

Optimization stop condition

Max iterations without progress. The maximum number of iterations without progress to perform before stopping the optimization, used after 250 initial iterations with early exaggeration. Note that progress is only checked every 50 iterations, so this value is rounded to the next multiple of 50. Default is 300, and the range is 0 - 9999999.

Min gradient norm. If the gradient norm is below this minimum threshold, the optimization will stop. Default is 1.0E-7.

Metric. The metric to use when calculating distance between instances in a feature array. If the metric is a string, it must be one of the options allowed by `scipy.spatial.distance.pdist` for its metric parameter, or a metric listed in `pairwise.PAIRWISE_DISTANCE_FUNCTIONS`. Select one of the available metric types. Default is euclidean.

When number of records greater than. Specify a method for plotting large datasets. You can specify a maximum dataset size or use the default 2,000 points. Performance is enhanced for large datasets when you select the Bin or Sample options. Alternatively, you can choose to plot all data points by selecting Use all data, but you should note that this may dramatically decrease the performance of the software.

- Bin. Select to enable binning when the dataset contains more than the specified number of records. Binning divides the graph into fine grids before actually plotting and counts the number of connections that would appear in each of the grid cells. In the final graph, one connection is used per cell at the bin centroid (average of all connection points in the bin).
- Sample. Select to randomly sample the data to the specified number of records.

The following table shows the relationship between the settings on the Expert tab of the SPSS® Modeler t-SNE node dialog and the Python t-SNE library parameters.

Table 1. Node properties mapped to Python library parameters

SPSS Modeler setting	Script name (property name)	Python t-SNE parameter
Mode	<code>mode_type</code>	
Visualization type	<code>n_components</code>	<code>n_components</code>
Method	<code>method</code>	<code>method</code>
Initialization of embedding	<code>init</code>	<code>init</code>
Target	<code>target_field</code>	<code>target_field</code>
Perplexity	<code>perplexity</code>	<code>perplexity</code>
Early exaggeration	<code>early_exaggeration</code>	<code>early_exaggeration</code>

SPSS Modeler setting	Script name (property name)	Python t-SNE parameter
Learning rate	learning_rate	learning_rate
Max iterations	n_iter	n_iter
Angular size	angle	angle
Set random seed	enable_random_seed	
Random seed	random_seed	random_state
Max iterations without progress	n_iter_without_progress	n_iter_without_progress
Min gradient norm	min_grad_norm	min_grad_norm
Perform t-SNE with multiple perplexities	isGridSearch	

t-SNE node Output options

Specify options for the t-SNE node output on the Output tab.

Output name. Specify the name of the output that is produced when the node runs. If you select Auto, the name of the output is automatically set.

Output to screen. Select this option to generate and display the output in a new window. The output is also added to the Output manager.

Output to file. Select this option to save the output to a file. Doing so enables the File name and File type fields. The t-SNE node requires access to this output file if you want to create plots using other fields for comparison purposes – or to use its output as predictors in classification or regression models. The t-SNE model creates a result file of x, y (and z) coordinate fields that is most easily accessed using a Fixed File source node.

t-SNE model nuggets

t-SNE model nuggets contain all of the information captured by the t-SNE model. The following tabs are available.

Graph

The Graph tab displays chart output for the t-SNE node. A pyplot scatter chart shows the low dimensions result. If you didn't select the Perform t-SNE with multiple perplexities option on the [Expert](#) tab of the t-SNE node, only one graph is included rather than six graphs with different perplexities.

Text output

The Text output tab displays the results of the t-SNE algorithm. If you chose the 2D visualization type on the [Expert](#) tab of the t-SNE node, the result here is the point value in two dimensions. If you chose 3D, the result is the point value in three dimensions.

Gaussian Mixture node

A Gaussian Mixture© model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians.¹

The Gaussian Mixture node in SPSS® Modeler exposes the core features and commonly used parameters of the Gaussian Mixture library. The node is implemented in Python.

For more information about Gaussian Mixture modeling algorithms and parameters, see the Gaussian Mixture documentation available at <http://scikit-learn.org/stable/modules/mixture.html> and <https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html>.²

¹ "User Guide." *Gaussian mixture models*. Web. © 2007 - 2017. scikit-learn developers.

² [Scikit-learn: Machine Learning in Python](#), Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

- [Gaussian Mixture node Fields](#)
- [Gaussian Mixture node Build Options](#)
- [Gaussian Mixture node Model Options](#)

Gaussian Mixture node Fields

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. This option uses the input settings from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign inputs, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the Predictors list on the right of the screen. The icons indicate the valid measurement levels for each field. To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Predictors. Select one or more fields as predictors.

Gaussian Mixture node Build Options

Use the Build Options tab to specify build options for the Gaussian Mixture node, including basic options and advanced options. For details about these options not covered in this section, see the following online resources:

- [Gaussian mixture parameter reference](#)¹
- [Gaussian mixture node user guide](#)²

Basic

Covariance type. Select one of the following covariance matrices:

- Full. Each component has its own general covariance matrix.
- Tied. All components share the same general covariance matrix.
- Diag. Each component has its own diagonal covariance matrix.
- Spherical. Each component has its own single variance.

Number of components. Specify the number of mixture components to use when building the model.

Cluster Label. Specify whether the cluster label is a number or a string. If you choose String, specify a prefix for the cluster label (for example, the default prefix is `cluster`, which results in cluster labels such as `cluster-1`, `cluster-2`, etc.).

Random Seed. Select this option and click Generate to generate the seed used by the random number generator.

Advanced

Tolerance. Specify the convergence threshold. Default value is 0.001.

Number of iterations. Specify the maximum number of iterations to perform. Default value is 100.

Init parameters. Select the initialization parameter Kmeans (responsibilities are initialized using k-means) or Random (responsibilities are initialized randomly).

Warm start. If you select True, the solution of the last fitting will be used as the initialization for the next fitting. This can speed up convergence when fitting is called several times on similar problems.

The following table shows the relationship between the settings in the SPSS® Modeler Gaussian Mixture node dialog and the Python Gaussian Mixture library parameters.

Table 1. Node properties mapped to Python library parameters

SPSS Modeler setting	Script name (property name)	Gaussian Mixture parameter
Use predefined roles / Use custom field assignments	<code>role_use</code>	
Inputs	<code>predictors</code>	
Use partitioned data	<code>use_partition</code>	
Covariance type	<code>covariance_type</code>	<code>covariance_type</code>
Number of components	<code>number_component</code>	<code>n_components</code>
Cluster Label	<code>component_label</code>	
Label Prefix	<code>label_prefix</code>	
Set random seed	<code>enable_random_seed</code>	
Random Seed	<code>random_seed</code>	<code>random_state</code>

SPSS Modeler setting	Script name (property name)	Gaussian Mixture parameter
Tolerance	<code>tol</code>	<code>tol</code>
Number of iterations	<code>max_iter</code>	<code>max_iter</code>
Init parameters	<code>init_params</code>	<code>init_params</code>
Warm start	<code>warm_start</code>	<code>warm_start</code>

¹ [Scikit-learn: Machine Learning in Python](#), Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

² ["User Guide."](#) Gaussian mixture models. Web. © 2007 - 2017. scikit-learn developers.

Gaussian Mixture node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

KDE nodes

Kernel Density Estimation (KDE)© uses the Ball Tree or KD Tree algorithms for efficient queries, and walks the line between unsupervised learning, feature engineering, and data modeling. Neighbor-based approaches such as KDE are some of the most popular and useful density estimation techniques. KDE can be performed in any number of dimensions, though in practice high dimensionality can cause a degradation of performance. The KDE Modeling and KDE Simulation nodes in SPSS® Modeler expose the core features and commonly used parameters of the KDE library. The nodes are implemented in Python.¹

To use a KDE node, you must set up an upstream Type node. The KDE node will read input values from the Type node (or the Types tab of an upstream source node).

The KDE Modeling node is available on SPSS Modeler's Modeling tab and Python tab. The KDE Modeling node generates a model nugget, and the nugget's scored values are kernel density values from the input data.

The KDE Simulation node is available on the Output tab and the Python tab. The KDE Simulation node generates a KDE Gen source node that can create some records that have the same distribution as the input data. The KDE Gen node includes a Settings tab where you can specify how many records the node will create (default is 1) and generate a random seed.

For more information about KDE, including examples, see the KDE documentation available at <http://scikit-learn.org/stable/modules/density.html#kernel-density-estimation>.¹

¹ "User Guide." Kernel Density Estimation. Web. © 2007-2018, scikit-learn developers.

- [KDE Modeling node and KDE Simulation node Fields](#)
- [KDE nodes Build Options](#)
- [KDE Modeling node and KDE Simulation node Model Options](#)
- [KDE Modeling node and KDE Simulation node Fields](#)
- [KDE nodes Build Options](#)
- [KDE Modeling node and KDE Simulation node Model Options](#)

KDE Modeling node and KDE Simulation node Fields

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. This option uses the input settings from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign inputs, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the Inputs list on the right of the screen. The icons indicate the valid measurement levels for each field. To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Inputs. Select one or more fields as inputs for clustering. KDE can only deal with continuous fields.

KDE nodes Build Options

Use the Build Options tab to specify build options for the KDE nodes, including basic options for kernel density parameters and cluster labels, and advanced options such as tolerance, leaf size, and whether to use a breadth-first approach. For additional information about these options, see the following online resources:

- [Kernel Density Estimation Python API Parameter Reference](#)¹
- [Kernel Density Estimation User Guide](#)²

Basic

Bandwidth. Specify the bandwidth of the kernel.

Kernel. Select the kernel to use. Available kernels for the KDE Modeling node are Gaussian, Tophat, Epanechnikov, Exponential, Linear, or Cosine. Available kernels for the KDE Simulation node are Gaussian or Tophat. For details about these available kernels, see the [Kernel Density Estimation User Guide](#).²

Algorithm. Select Auto, Ball Tree or KD Tree for the tree algorithm to use. For more information, see [Ball Tree](#)³ and [KD Tree](#).⁴

Metric. Select a distance metric. Available metrics are Euclidean, Braycurtis, Chebyshev, Canberra, Cityblock, Dice, Hamming, Infinity, Jaccard, L1, L2, Matching, Manhattan, P, Rogerstanimoto, Russellrao, Sokalmichener, Sokalsneath, Kulsinski, or Minkowski. If you select Minkowski, set the P Value as desired.

The metrics available in this drop-down will vary depending on which algorithm you choose. Also note that the normalization of the density output is correct only for the Euclidean distance metric.

Advanced

Absolute Tolerance. Specify the desired absolute tolerance of the result. A larger tolerance will generally result in faster run time. Default is 0.0.

Relative Tolerance. Specify the desired relative tolerance of the result. A larger tolerance will generally result in faster run time. Default is 1E-8.

Leaf Size. Specify the leaf size of the underlying tree. Default is 40. Changing the leaf size may significantly impact performance and required memory. For more information about the Ball Tree and KD Tree algorithms, see [Ball Tree](#)³ and [KD Tree](#).⁴

Breadth first. Select True if you want to use a breadth-first approach or False to use a depth-first approach.

The following table shows the relationship between the settings in the SPSS® Modeler KDE node dialogs and the Python KDE library parameters.

Table 1. Node properties mapped to Python library parameters

SPSS Modeler setting	Script name (property name)	KDE parameter
Inputs	inputs	
Bandwidth	bandwidth	bandwidth
Kernel	kernel	kernel
Algorithm	algorithm	algorithm
Metric	metric	metric
P Value	pValue	pValue
Absolute Tolerance	atol	atol
Relative Tolerance	rtol	Rtol
Leaf Size	leafSize	leafSize
Breadth first	breadthFirst	breadthFirst

¹ "API Reference." *sklearn.neighbors.KernelDensity*. Web. © 2007-2018, scikit-learn developers.

² "User Guide." *Kernel Density Estimation*. Web. © 2007-2018, scikit-learn developers.

³ [Ball Tree](#). "Five balltree construction algorithms. © 1989, Omohundro, S.M., International Computer Science Institute Technical Report.

⁴ [K-D Tree](#). "Multidimensional binary search trees used for associative searching. © 1975, Bentley, J.L., Communications of the ACM.

KDE Modeling node and KDE Simulation node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Random Forest node

Random Forest® is an advanced implementation of a bagging algorithm with a tree model as the base model. In random forests, each tree in the ensemble is built from a sample drawn with replacement (for example, a bootstrap sample) from the training set. When splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. Because of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree) but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model.¹

The Random Forest node in SPSS® Modeler is implemented in Python. The Python tab on the Nodes Palette contains this node and other Python nodes.

For more information about random forest algorithms, see <https://scikit-learn.org/stable/modules/ensemble.html#forest>.

¹L. Breiman, "Random Forests," *Machine Learning*, 45(1), 5-32, 2001.

- [Random Forest node Fields](#)
- [Random Forest node Build Options](#)
- [Random Forest node Model Options](#)
- [Random Forest model nuggets](#)

Random Forest node Fields

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign a target and predictors, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the Target and Predictors role fields on the right of the screen. The icons indicate the valid measurement levels for each role field. To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Target. Select a field to use as the target for the prediction.

Predictors. Select one or more fields as inputs for the prediction.

Random Forest node Build Options

Use the Build Options tab to specify build options for the Random Forest node, including basic options and advanced options. For more information about these options, see <https://scikit-learn.org/stable/modules/ensemble.html#forest>

Basic

Number of trees to build. Select the number of trees in the forest.

Specify max depth. If not selected, nodes are expanded until all leaves are pure or until all leaves contain less than `min_samples_split` samples.

Max depth. The maximum depth of the tree.

Minimum leaf node size. The minimum number of samples required to be at a leaf node.

Number of features to use for splitting. The number of features to consider when looking for the best split:

- If `auto`, then `max_features=sqrt(n_features)` for classifier and `max_features=n_features` for regression.
- If `sqrt`, then `max_features=sqrt(n_features)`.
- If `log2`, then `max_features=log2(n_features)`.

Advanced

Use bootstrap samples when building trees. If selected, bootstrap samples are used when building trees.

Use out-of-bag samples to estimate the generalization accuracy. If selected, out-of-bag samples are used to estimate the generalization accuracy.

Use extremely randomized trees. If selected, extremely randomized trees are used instead of general random forests. In extremely randomized trees, randomness goes one step further in the way splits are computed. As in random forests, a random subset of candidate features is used, but instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds is picked as the splitting rule. This usually allows the variance of the model to be reduced a bit more, at the expense of a slightly greater increase in bias.¹

Replicate results. If selected, the model building process is replicated to achieve the same scoring results.

Random seed. You can click Generate to generate the seed used by the random number generator.

Hyper-Parameter Optimization (Based on Rbfopt). Select this option to enable Hyper-Parameter Optimization based on Rbfopt, which automatically discovers the optimal combination of parameters so that the model will achieve the expected or lower error rate on the samples. For details about Rbfopt, see http://rbfopt.readthedocs.io/en/latest/rbfopt_settings.html.

Target. The objective function value (error rate of the model on the samples) you want to reach (i.e., the value of the unknown optimum). Set to an acceptable value such as 0.01.

Max iterations. The maximum number of iterations to try the model. Default is 1000.

Max evaluations. The maximum number of function evaluations in accurate mode, for trying the model. Default is 300.

The following table shows the relationship between the settings in the SPSS® Modeler Random Forest node dialog and the Python Random Forest library parameters.

Table 1. Node properties mapped to Python library parameters

SPSS Modeler setting	Script name (property name)	Random Forest parameter
Target	<code>target</code>	
Predictors	<code>inputs</code>	
Number of trees to build	<code>n_estimators</code>	<code>n_estimators</code>
Specify max depth	<code>specify_max_depth</code>	<code>specify_max_depth</code>
Max depth	<code>max_depth</code>	<code>max_depth</code>
Minimum leaf node size	<code>min_samples_leaf</code>	<code>min_samples_leaf</code>
Number of features to use for splitting	<code>max_features</code>	<code>max_features</code>
Use bootstrap samples when building trees	<code>bootstrap</code>	<code>bootstrap</code>
Use out-of-bag samples to estimate the generalization accuracy	<code>oob_score</code>	<code>oob_score</code>
Use extremely randomized trees	<code>extreme</code>	
Replicate results	<code>use_random_seed</code>	
Random seed	<code>random_seed</code>	<code>random_seed</code>
Hyper-Parameter Optimization (based on Rbfopt)	<code>enable_hpo</code>	
Target (for HPO)	<code>target_objval</code>	
Max iterations (for HPO)	<code>max_iterations</code>	
Max evaluations (for HPO)	<code>max_evaluations</code>	

¹L. Breiman, "Random Forests," Machine Learning, 45(1), 5-32, 2001.

Random Forest node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Random Forest model nuggets

Random Forest model nuggets contain all of the information captured by the random forest model. The following sections are available.

Model Information

This view provides key information about the model, including input fields, one-hot encoding values, and model parameters.

Predictor Importance

This view displays a chart that indicates the relative importance of each predictor in estimating the model. For more information, see [Predictor Importance](#).

HDBSCAN node

Hierarchical Density-Based Spatial Clustering (HDBSCAN)© uses unsupervised learning to find clusters, or dense regions, of a data set. The HDBSCAN node in SPSS® Modeler exposes the core features and commonly used parameters of the HDBSCAN library. The node is implemented in Python, and you can use it to cluster your dataset into distinct groups when you don't know what those groups are at first. Unlike most learning methods in SPSS Modeler, HDBSCAN models do *not* use a target field. This type of learning, with no target field, is called *unsupervised learning*. Rather than trying to predict an outcome, HDBSCAN tries to uncover patterns in the set of input fields. Records are grouped so that records within a group or cluster tend to be similar to each other, but records in different groups are dissimilar. The HDBSCAN algorithm views clusters as areas of high density separate by areas of low density. Due this rather generic view, clusters found by HDBSCAN can be any shape, as opposed to k-means which assumes that clusters are convex shaped. Outlier points that lie alone in low-density regions are also marked. HDBSCAN also supports scoring of new samples.¹

To use the HDBSCAN node, you must set up an upstream Type node. The HDBSCAN node will read input values from the Type node (or the Types tab of an upstream source node).

For more information about HDBSCAN clustering algorithms, see the HDBSCAN documentation available at <http://hdbSCAN.readthedocs.io/en/latest/>.¹

¹ "User Guide / Tutorial." *The hdbSCAN Clustering Library*. Web. © 2016, Leland McInnes, John Healy, Steve Astels.

- [HDBSCAN node Fields](#)
 - [HDBSCAN node Build Options](#)
 - [HDBSCAN node Model Options](#)
-

HDBSCAN node Fields

The Fields tab specifies which fields are used in the analysis.

Important: To train an HDBSCAN model, you must use one or more fields with the role set to Input. Fields with the role set to Output, Both, or None are ignored.

Use predefined roles. This option uses the input settings from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign inputs, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the Inputs list on the right of the screen. The icons indicate the valid measurement levels for each field. To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Inputs. Select one or more fields as inputs for clustering.

HDBSCAN node Build Options

Use the Build Options tab to specify build options for the HDBSCAN node, including basic options for cluster parameters and cluster labels, and advanced options for advanced parameters and chart output options. For additional information about these options, see the following online resources:

- [HDBSCAN Python API Parameter Reference](#)¹
- [HDBSCAN Home Page](#)²

Basic

Hyper-Parameter Optimization (Based on Rbfopt). Select this option to enable Hyper-Parameter Optimization based on Rbfopt, which automatically discovers the optimal combination of parameters so that the model will achieve the expected or lower error rate on the samples. For details about Rbfopt, see http://rbfopt.readthedocs.io/en/latest/rbfopt_settings.html.

Min Cluster Size. Specify the minimum size of clusters. Single linkage splits that contain fewer points than the value specified here will be considered points "falling out" of a cluster rather than a cluster splitting into two new clusters.

Min Samples. Specify the minimum number of samples in a neighborhood for a point to be considered a core point. If set to 0, the default value is the minimum cluster size value.

Algorithm. Select the algorithm to use. HDBSCAN has variants that are specialized for different characteristics of the data. By default, BEST is used - which automatically chooses the best algorithm given the nature of the data. For details about these algorithm types, see the [HDBSCAN documentation](#).¹ Note that the algorithm you choose will impact performance. For example, for large data we recommend trying Boruvka KDTree or Boruvka BallTree.

Metric for Distance. Select the metric to use when calculating distance between instances in a feature array.

Cluster Label. Specify whether the cluster label is a number or a string. If you choose String, specify a prefix for the cluster label (for example, the default prefix is `cluster`, which results in cluster labels such as `cluster-1`, `cluster-2`, etc.).

Advanced

Approximate Minimum Spanning Tree. Select True if you want to accept an approximate minimum spanning tree. For some algorithms, this can improve performance, but the resulting clustering might be of marginally lower quality. If you are willing to sacrifice speed for correctness, you may want to try the False option. In most cases, True is recommended.

Method to Select Cluster. Select which method to use for selecting clusters from the condensed tree. The standard approach for HDBSCAN is to use an Excess of Mass (EOM) algorithm to find the most persistent clusters. Or you can select the clusters at the leaves of the tree, which provides the most fine-grained and homogeneous clusters.

Accept Single Cluster. Change this setting to True to allow single cluster results only if this is a valid result for your dataset.

P Value. If using the Minkowski metric for distance (under Basic build options), you can change this p value if desired.

Leaf Size. If using a space tree algorithm (Boruvka KDTree or Boruvka BallTree), this is the number of points in a leaf node of the tree. This setting doesn't alter the resulting clustering, but it may impact the run time of the algorithm.

Validity Index. Select this option to include the Validity Index chart in the model nugget output.

Condensed Tree. Select this option to include the Condensed Tree chart in the model nugget output.

Single Linkage Tree. Select this option to include the Single Linkage Tree chart in the model nugget output.

Min Span Tree. Select this option to include the Min Span Tree chart in the model nugget output.

The following table shows the relationship between the settings in the SPSS® Modeler HDBSCAN node dialog and the Python HDBSCAN library parameters.

Table 1. Node properties mapped to Python library parameters

SPSS Modeler setting	Script name (property name)	HDBSCAN parameter
Inputs	<code>inputs</code>	<code>inputs</code>
Hyper-Parameter Optimization	<code>useHPO</code>	
Min Cluster Size	<code>min_cluster_size</code>	<code>min_cluster_size</code>
Min Samples	<code>min_samples</code>	<code>min_samples</code>
Algorithm	<code>algorithm</code>	<code>algorithm</code>
Metric for Distance	<code>metric</code>	<code>metric</code>
Cluster Label	<code>useStringLabel</code>	
Label Prefix	<code>stringLabelPrefix</code>	
Approximate Minimum Spanning Tree	<code>approx_min_span_tree</code>	<code>approx_min_span_tree</code>
Method to Select Cluster	<code>cluster_selection_method</code>	<code>cluster_selection_method</code>
Accept Single Cluster	<code>allow_single_cluster</code>	<code>allow_single_cluster</code>
P Value	<code>p_value</code>	<code>p_value</code>
Leaf Size	<code>leaf_size</code>	<code>leaf_size</code>
Validity Index	<code>outputValidity</code>	
Condensed Tree	<code>outputCondensed</code>	
Single Linkage Tree	<code>outputSingleLinkage</code>	
Min Span Tree	<code>outputMinSpan</code>	

¹ "API Reference." *The hdbscan Clustering Library*. Web. © 2016, Leland McInnes, John Healy, Steve Astels.

² "User Guide / Tutorial." *The hdbscan Clustering Library*. Web. © 2016, Leland McInnes, John Healy, Steve Astels.

HDBSCAN node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

One-Class SVM node

The One-Class SVM[©] node uses an unsupervised learning algorithm. The node can be used for novelty detection. It will detect the soft boundary of a given set of samples, to then classify new points as belonging to that set or not. This One-Class SVM modeling node is implemented in Python and requires the scikit-learn[©] Python library. For details about the scikit-learn library, see <http://contrib.scikit-learn.org/imbalanced-learn/about.html>¹.

The Python tab on the Nodes Palette contains the One-Class SVM node and other Python nodes.

Note: One-Class SVM is used for unsupervised outlier and novelty detection. In most cases, we recommend using a known, "normal" dataset to build the model so the algorithm can set a correct boundary for the given samples. Parameters for the model – such as nu, gamma, and kernel – impact the result significantly. So you may need to experiment with these options until you find the optimal settings for your situation.

¹Smola, Schölkopf. "A Tutorial on Support Vector Regression." *Statistics and Computing Archive*, vol. 14, no. 3, August 2004, pp. 199-222. (<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.114.4288>)

- [One-Class SVM node Fields](#)
- [One-Class SVM node Expert](#)
- [One-Class SVM node Options](#)

One-Class SVM node Fields

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. Select this option to select all fields with a defined role of Input.

Use custom field assignments. To manually select fields, select this option and choose input fields and split fields:

Inputs. Select the input fields to use in the analysis. All storage types and measurement types are supported, except for typeless or unknown. If a field has a String storage type, the values of this field will be binarized in a one-vs-all fashion via a one-hot encoding algorithm.

Split. Select which field or fields to use as split fields. All flag, nominal, ordinal, and discrete measurement types are supported.

Use partitioned data If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

One-Class SVM node Expert

On the Expert tab of the One-Class SVM node, you can choose from Simple mode or Expert mode. If you choose Simple, all parameters are set with the default values as shown below. If you select Expert, you can specify custom values for these parameters. For further detail about these options, see <http://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html#sklearn.svm.OneClassSVM>.

Stopping criteria. Specify the tolerance for stopping criteria. Default is 1.0E-3 (0.001).

Regression precision (nu). Bound on the fraction of training errors and support vectors. Default is 0.1.

Kernel type. The kernel type to use in the algorithm. Options include RBF, Polynomial, Sigmoid, Linear, or Precomputed. Default is RBF.

Specify Gamma. Select this option to specify the Gamma. Otherwise, the auto gamma will be applied.

Gamma. The Gamma setting is only available for the RBF, Polynomial, and Sigmoid kernel types.

Coef0. Coef0 is only available for the Polynomial and Sigmoid kernel types.

Degree. Degree is only available for the Polynomial kernel type.

Use the shrinking heuristic. Select this option to use the shrinking heuristic. This option is deselected by default.

Specify the size of the kernel cache (in MB). Select this option to specify the size of the kernel cache. This option is deselected by default. When selected, the default value is 200 MB.

Hyper-Parameter Optimization (Based on Rbfopt). Select this option to enable Hyper-Parameter Optimization based on Rbfopt, which automatically discovers the optimal combination of parameters so that the model will achieve the expected or lower error rate on the samples. For details about Rbfopt, see http://rbfopt.readthedocs.io/en/latest/rbfopt_settings.html.

Target. The objective function value (error rate of the model on the samples) we want to reach (for example, the value of the unknown optimum). Set to an acceptable value such as 0.01.

Max Iterations. Maximum number of iterations for trying the model. Default is 1000.

Max Evaluations. Maximum number of function evaluations for trying the model, where the focus is accuracy over speed. Default is 300.

The One-Class SVM node requires the scikit-learn® Python library. The following table shows the relationship between the settings in the SPSS® Modeler SMOTE node dialog and the Python algorithm.

Table 1. Node properties mapped to Python library parameters

Parameter name	Script name (property name)	Python API parameter name
Stopping criteria	<code>stopping_criteria</code>	<code>tol</code>
Regression precision	<code>precision</code>	<code>nu</code>
Kernel type	<code>kernel</code>	<code>kernel</code>
Gamma	<code>gamma</code>	<code>gamma</code>
Coef0	<code>coef0</code>	<code>coef0</code>
Degree	<code>degree</code>	<code>degree</code>
Use the shrinking heuristic	<code>shrinking</code>	<code>shrinking</code>
Specify the size of the kernel cache (number input box)	<code>cache_size</code>	<code>cache_size</code>
Random seed	<code>random_seed</code>	<code>random_state</code>

One-Class SVM node Options

On the Options tab of the One-Class SVM node, you can set the following options.

Type of Parallel Coordinates graphic. SPSS® Modeler draws parallel coordinates graphics to present the built model. Sometimes, the values for some data columns/features will be displayed much larger than others, which can make some other portions of the graph hard to see. For cases like this, you can choose the Independent vertical axes option to give all vertical axes a standalone axis scale, or select General vertical axes to force all vertical axes to share the same axes scale.

Maximum lines on the graphic. Specify the maximum number of data rows (lines) to display in graph output. The default is 100. For performance reasons, a maximum of 20 fields will be displayed.

Draw all input fields on the graphic. Select this option to show all input fields in the graph output. By default, each data field will be drawn as a vertical axis. For performance reasons, a maximum of 30 fields will be displayed.

Custom fields to be drawn on the graphic. Rather than showing all input fields in the graph output, you can select this option and choose a subset of fields to show. This can improve performance. For performance reasons, a maximum of 20 fields will be displayed.

Spark nodes

SPSS® Modeler offers nodes for using Spark native algorithms. The Spark tab on the [Nodes Palette](#) contains the following nodes you can use to run Spark algorithms. These nodes are supported on Windows 64, Mac 64, and Linux 64. Note that these nodes don't support specifying an integer/double column as Flag/Nominal for building a model. To do this, you must convert the column value to 0/1 or 0,1,2,3,4...

	Isotonic Regression belongs to the family of regression algorithms. The Isotonic-AS node in SPSS Modeler is implemented in Spark. For details about Isotonic Regression algorithms, see https://spark.apache.org/docs/2.2.0/mllib-isotonic-regression.html .
	XGBoost® is an advanced implementation of a gradient boosting algorithm. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost-AS node in SPSS Modeler exposes the core features and commonly used parameters. The XGBoost-AS node is implemented in Spark.
	K-Means is one of the most commonly used clustering algorithms. It clusters data points into a predefined number of clusters. The K-Means-AS node in SPSS Modeler is implemented in Spark. For details about K-Means algorithms, see https://spark.apache.org/docs/2.2.0/ml-clustering.html . Note that the K-Means-AS node performs one-hot encoding automatically for categorical variables.
	Multilayer perceptron is a classifier based on the feedforward artificial neural network and consists of multiple layers. Each layer is fully connected to the next layer in the network. The MultiLayerPerceptron-AS node in SPSS Modeler is implemented in Spark. For details about the multilayer perceptron classifier (MLPC), see https://spark.apache.org/docs/latest/ml-classification-regression.html#multilayer-perceptron-classifier .

- [Isotonic-AS node](#)

- [XGBoost-AS node](#)
 - [K-Means-AS node](#)
 - [MultiLayerPerceptron-AS node](#)
-

Isotonic-AS node

Isotonic Regression belongs to the family of regression algorithms. The Isotonic-AS node in SPSS® Modeler is implemented in Spark.

For details about Isotonic Regression algorithms, see <https://spark.apache.org/docs/2.2.0/mllib-isotonic-regression.html>.¹

¹ "Regression - RDD-based API." *Apache Spark. MLLib: Main Guide*. Web. 3 Oct 2017.

- [Isotonic-AS node Fields](#)
 - [Isotonic-AS node Build Options](#)
 - [Isotonic-AS model nuggets](#)
-

Isotonic-AS node Fields

The Fields tab specifies which fields are used in the analysis.

Fields. Lists all fields in the data source. Use the arrow buttons to assign items manually from this list to the Target, Input, and Weight fields on the right of the screen. The icons indicate the valid measurement levels for each role field. To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Target. Select a field to use as the target.

Input. Select the input field or fields.

Weight. Select a weight field for exponential weight. If not set, the default weight value of 1 will be used.

Isotonic-AS node Build Options

Use the Build Options tab to specify build options for the Isotonic-AS node, including the feature index and isotonic type. For more information, see <http://spark.apache.org/docs/latest/api/java/org/apache/spark/ml/regression/IsotonicRegression.html>.¹

Input Fields Index. Specify the index of the input fields. Default is 0.

Isotonic Type. This setting determines whether the output sequence should be isotonic/increasing or antitonic/decreasing. Default is Isotonic.

¹ "Class IsotonicRegression." *Apache Spark. JavaDoc*. Web. 3 Oct 2017.

Isotonic-AS model nuggets

Isotonic-AS model nuggets contain all of the information captured by the isotonic regression model. The following sections are available.

Model Summary

This view provides key information about the model, including input fields, target field, and model building options.

Model Chart

This view displays a scatter diagram.

XGBoost-AS node

XGBoost® is an advanced implementation of a gradient boosting algorithm. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost-AS node in SPSS® Modeler exposes the core features and commonly used parameters. The XGBoost-AS node is implemented in Spark.

For more information about boosting algorithms, see the XGBoost Tutorials available at <http://xgboost.readthedocs.io/en/latest/tutorials/index.html>.¹

Note that the XGBoost cross-validation function is not supported in SPSS Modeler. You can use the SPSS Modeler Partition node for this functionality. Also note that XGBoost in SPSS Modeler performs one-hot encoding automatically for categorical variables.

Note: On Mac, version 10.12.3 or higher is required for building XGBoost-AS models.

¹ "XGBoost Tutorials." *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

- [XGBoost-AS node Fields](#)
 - [XGBoost-AS node Build Options](#)
 - [XGBoost-AS node Model Options](#)
-

XGBoost-AS node Fields

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign a target and predictors, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the Target and Predictors role fields on the right of the screen. The icons indicate the valid measurement levels for each role field. To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Target. Select a field to use as the target for the prediction.

Predictors. Select one or more fields as inputs for the prediction.

XGBoost-AS node Build Options

Use the Build Options tab to specify build options for the XGBoost-AS node, including general options for model building and handling imbalanced datasets, learning task options for objectives and evaluation metrics, and booster parameters for specific boosters. For more information about these options, see the following online resources:

- [XGBoost Home Page](#)¹
- [XGBoost Parameter Reference](#)²
- [XGBoost Spark API](#)³

General

Number of Workers. Number of workers used to train the XGBoost model.

Number of Threads. Number of threads used per worker.

Use External Memory. Whether to use external memory as cache.

Booster Type. The booster to use (gbtree, gblinear, or dart).

Booster Rounds Number. The number of rounds for boosting.

Scale pos weight. This setting controls the balance of positive and negative weights, and is useful for unbalanced classes.

Random Seed. Click Generate to generate the seed used by the random number generator.

Learning Task

Objective. Select from the following learning task objective types: reg:linear, reg:logistic, reg:gamma, reg:tweedie, rank:pairwise, binary:logistic, or multi.

Evaluation Metrics. Evaluation metrics for validation data. A default metric will be assigned according to the objective (rmse for regression, error for classification, or mean average precision for ranking). Available options are rmse, mae, logloss, error, merror, mlogloss, uac, ndcg, map, or gamma-deviance (default is rmse).

Booster Parameters

Lambda. L2 regularization term on weights. Increasing this value will make the model more conservative.

Alpha. L1 regularization term on weights. Increasing this value will make model more conservative.

Lambda bias. L2 regularization term on bias. (There is no L1 regularization term on bias because it is not important.)

Tree method. Select the XGBoost tree construction algorithm to use.

Max depth. Specify the maximum depth for trees. Increasing this value will make the model more complex and likely to be overfitting.

Min child weight. Specify the minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than this Min child weight, then the building process will stop further partitioning. In linear regression mode, this simply corresponds to minimum number of instances needed in each node. The larger the weight, the more conservative the algorithm will be.

Max delta step. Specify the maximum delta step to allow for each tree's weight estimation. If set to 0, there is no constraint. If set to a positive value, it can help the update step be more conservative. Usually this parameter is not needed, but it may help in logistic regression when a class is extremely imbalanced.

Sub sample. Sub sample is the ratio of the training instance. For example, if you set this to 0.5, XGBoost will randomly collect half the data instances to grow trees and this will prevent overfitting.

Eta. The step size shrinkage used during the update step to prevent overfitting. After each boosting step, the weights of new features can be obtained directly. Eta also shrinks the feature weights to make the boosting process more conservative.

Gamma. The minimum loss reduction required to make a further partition on a leaf node of the tree. The larger the gamma setting, the more conservative the algorithm will be.

Colsample by tree. Sub sample ratio of columns when constructing each tree.

Colsample by level. Sub sample ratio of columns for each split, in each level.

Normalization Algorithm. The normalization algorithm to use when the dart booster type is selected under General options. Available options are tree or forest (default is tree).

Sampling Algorithm. The sampling algorithm to use when the dart booster type is selected under General options. The uniform algorithm uniformly selects dropped trees. The weighted algorithm selects dropped trees in proportion to weight. The default is uniform.

Dropout Rate. The dropout rate to use when the dart booster type is selected under General options.

Probability of Skip Dropout. The skip dropout probability to use when the dart booster type is selected under General options. If a dropout is skipped, new trees are added in the same manner as gbtree.

The following table shows the relationship between the settings in the SPSS® Modeler XGBoost-AS node dialog and the XGBoost Spark parameters.

Table 1. Node properties mapped to Spark parameters

SPSS Modeler setting	Script name (property name)	XGBoost Spark parameter
Target	target_fields	
Predictors	input_fields	
Lambda	lambda	lambda
Number of Workers	nWorkers	nWorkers
Number of Threads	numThreadPerTask	numThreadPerTask
Use External Memory	useExternalMemory	useExternalMemory
Booster Type	boosterType	boosterType
Boosting Round Number	numBoostRound	round
Scale Pos Weight	scalePosWeight	scalePosWeight
Objective	objectiveType	objective
Evaluation Metrics	evalMetric	evalMetric
Lambda	lambda	lambda
Alpha	alpha	alpha
Lambda bias	lambdaBias	lambdaBias
Tree Method	treeMethod	treeMethod
Max Depth	maxDepth	maxDepth

SPSS Modeler setting	Script name (property name)	XGBoost Spark parameter
Min child weight	minChildWeight	minChildWeight
Max delta step	maxDeltaStep	maxDeltaStep
Sub sample	sampleSize	sampleSize
Eta	eta	eta
Gamma	gamma	gamma
Colsample by tree	colsSampleRation	colSampleByTree
Colsample by level	colsSampleLevel	colSampleLevel
Normalization Algorithm	normalizeType	normalizeType
Sampling Algorithm	sampleType	sampleType
Dropout Rate	rateDrop	rateDrop
Probability of Skip Dropout	skipDrop	skipDrop

¹ "Scalable and Flexible Gradient Boosting." Web. © 2015-2016 DMLC.

² "XGBoost Parameters" *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

³ "ml.dmlc.xgboost4j.scala.spark Params." *DMLC for Scalable and Reliable Machine Learning*. Web. 3 Oct 2017.

XGBoost-AS node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

K-Means-AS node

K-Means is one of the most commonly used clustering algorithms. It clusters data points into a predefined number of clusters.¹ The K-Means-AS node in SPSS® Modeler is implemented in Spark.

For details about K-Means algorithms, see <https://spark.apache.org/docs/2.2.0/ml-clustering.html>.

Note that the K-Means-AS node performs one-hot encoding automatically for categorical variables.

¹ "Clustering." *Apache Spark*. MLlib: Main Guide. Web. 3 Oct 2017.

- [K-Means-AS node Fields](#)
- [K-Means-AS node Build Options](#)
- [K-Means-AS node Fields](#)
- [K-Means-AS node Build Options](#)

K-Means-AS node Fields

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. This option tells the node to use field information from an upstream Type node. It's selected by default.

Use custom field assignments. If you want to manually assign input fields, select this option and then select the input field or fields. Using this option is similar to setting the field role in Input in a Type node.

K-Means-AS node Build Options

Use the Build Options tab to specify build options for the K-Means-AS node, including regular options for model building, initialization options for initializing cluster centers, and advanced options for the computing iteration and random seed. For more information, see the [JavaDoc for K-Means on SparkML](#).¹

Regular

Model Name. The name of the field generated after scoring to a specific cluster. Select Auto (default) or select Custom and type a name.

Number of Clusters. Specify the number of clusters to generate. The default is 5 and the minimum is 2.

Initialization

Initialization Mode. Specify the method for initializing the cluster centers. K-Means|| is the default. For details about these two methods, see [Scalable K-Means++](#).²

Initialization Steps. If the K-Means|| initialization mode is selected, specify the number of initialization steps. 2 is the default.

Advanced

Advanced Settings. Select this option if you want to set advanced options as follows.

Max Iteration. Specify the maximum number of iterations to perform when searching cluster centers. 20 is the default.

Tolerance. Specify the convergence tolerance for iterative algorithms. 1.0E-4 is the default.

Set Random Seed. Select this option and click Generate to generate the seed used by the random number generator.

Display

Display Graph. Select this option if you want a graph to be included in the output.

The following table shows the relationship between the settings in the SPSS® Modeler K-Means-AS node and the K-Means Spark parameters.

Table 1. Node properties mapped to Spark parameters

SPSS Modeler setting	Script name (property name)	K-Means SparkML parameter
Input Fields	features	
Number of Clusters	clustersNum	k
Initialization Mode	initMode	initMode
Initialization Steps	initSteps	initSteps
Max Iteration	maxIter	maxIter
Toleration	toleration	tol
Random Seed	randomSeed	seed

¹ "Class KMeans." *Apache Spark*. JavaDoc. Web. 3 Oct 2017.

² Bahmani, Moseley, et al. "Scalable K-Means++." Feb 28, 2012. <http://theory.stanford.edu/%7Esergei/papers/vldb12-kmpar.pdf>.

MultiLayerPerceptron-AS node

Multilayer perceptron is a classifier based on the feedforward artificial neural network and consists of multiple layers. Each layer is fully connected to the next layer in the network. For details about the multilayer perceptron classifier (MLPC), see

<https://spark.apache.org/docs/latest/ml-classification-regression.html#multilayer-perceptron-classifier>.¹

The MultiLayerPerceptron-AS node in SPSS® Modeler is implemented in Spark. To use a this node, you must set up an upstream Type node. The MultiLayerPerceptron-AS node will read input values from the Type node (or the Types tab of an upstream source node).

¹ "Multilayer perceptron classifier." *Apache Spark*. MLlib: Main Guide. Web. 5 Oct 2018.

- [MultiLayerPerceptron-AS node Fields](#)
- [MultiLayerPerceptron-AS node Build Options](#)
- [MultiLayerPerceptron node Model Options](#)

MultiLayerPerceptron-AS node Fields

The Fields tab specifies which fields are used in the analysis.

Use predefined roles. This option tells the node to use field information from an upstream Type node. This is the default.

Use custom field assignments. To manually assign target and predictors, select this option.

Target. Select a field to use as the target for the prediction.

Predictors. Select one or more fields to use as inputs for the prediction.

MultiLayerPerceptron-AS node Build Options

Use the Build Options tab to specify build options for the MultiLayerPerceptron-AS node, including performance, modeling building, and expert options. For more information about these options, see

<http://spark.apache.org/docs/latest/api/java/org/apache/spark/ml/classification/MultilayerPerceptronClassifier.html>.¹

Performance

Perceptrons Layer. Use this setting to define the number of perceptron layers to include. This value must be larger than the number of perceptron fields. The default value is 1.

Hidden Layers. Specify the number of hidden layers. Use a comma between multiple hidden layers. The default value is 1.

Output Layer. Specify the number of output layers. The default value is 1.

Random Seed. Click Generate if you want to generate the seed used by the random number generator.

Model Building

Max iterations. Specify the maximum number of iterations to perform. The default value is 10.

Expert Only

BlockSize. Select the Expert Mode option in the Model Building section if you want to specify the block size for stacking input data in matrices. This can speed up the computation. The default block size is 128.

The following table shows the relationship between the settings in the SPSS® Modeler MultiLayerPerceptron-AS node dialogs and the Spark KDE library parameters.

Table 1. Node properties mapped to Spark parameters

SPSS Modeler setting	Script name (property name)	Spark parameter
Predictors	<code>features</code>	
Target	<code>label</code>	
Perceptrons Layer	<code>layers[0]</code>	<code>layers[0]</code>
Hidden Layers	<code>layers[1...<latest-1>]</code>	<code>layers[1...<latest-1>]</code>
Output Layer	<code>layers[<latest>]</code>	<code>layers[<latest>]</code>
Random Seed	<code>seed</code>	<code>seed</code>
Max iterations	<code>maxiter</code>	<code>maxiter</code>

¹ "Class MultilayerPerceptronClassifier." Apache Spark. JavaDoc. Web. 5 Oct 2018.

MultiLayerPerceptron node Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Scripting and the Scripting Language

- [Scripting overview](#)
- [Types of Scripts](#)
- [Stream Scripts](#)
- [Standalone Scripts](#)
- [SuperNode Scripts](#)
- [Looping and conditional execution in streams](#)
- [Executing and interrupting scripts](#)

Scripting overview

Scripting in IBM® SPSS® Modeler is a powerful tool for automating processes in the user interface. Scripts can perform the same types of actions that you perform with a mouse or a keyboard, and you can use them to automate tasks that would be highly repetitive or time consuming to perform manually.

You can use scripts to:

- Impose a specific order for node executions in a stream.
- Set properties for a node as well as perform derivations using a subset of CLEM (Control Language for Expression Manipulation).
- Specify an automatic sequence of actions that normally involves user interaction--for example, you can build a model and then test it.
- Set up complex processes that require substantial user interaction--for example, cross-validation procedures that require repeated model generation and testing.
- Set up processes that manipulate streams—for example, you can take a model training stream, run it, and produce the corresponding model-testing stream automatically.

This section provides high-level descriptions and examples of stream-level scripts, standalone scripts, and scripts within SuperNodes in the IBM SPSS Modeler interface. More information on scripting language, syntax, and commands is provided in the sections that follow.

Note:

You cannot import and run scripts created in IBM SPSS Statistics within IBM SPSS Modeler.

Related information

- [Types of Scripts](#)
- [Stream Scripts](#)
- [Stream script example: Training a neural net](#)
- [Standalone Scripts](#)
- [SuperNode Scripts](#)
- [CLEM Reference Overview](#)

Types of Scripts

IBM® SPSS® Modeler uses three types of scripts:

- **Stream scripts** are stored as a stream property and are therefore saved and loaded with a specific stream. For example, you can write a stream script that automates the process of training and applying a model nugget. You can also specify that whenever a particular stream is executed, the script should be run instead of the stream's canvas content. See the topic [Stream Scripts](#) for more information.
- **Standalone scripts** are not associated with any particular stream and are saved in external text files. You might use a standalone script, for example, to manipulate multiple streams together. See the topic [Standalone Scripts](#) for more information.
- **SuperNode scripts** are stored as a SuperNode stream property. SuperNode scripts are only available in terminal SuperNodes. You might use a SuperNode script to control the execution sequence of the SuperNode contents. For nonterminal (source or process) SuperNodes, you can define properties for the SuperNode or the nodes it contains in your stream script directly. See the topic [SuperNode Scripts](#) for more information.

Related information

- [Scripting overview](#)
- [Stream Scripts](#)
- [Stream script example: Training a neural net](#)
- [Standalone Scripts](#)
- [SuperNode Scripts](#)

Stream Scripts

Scripts can be used to customize operations within a particular stream, and they are saved with that stream. Stream scripts can be used to specify a particular execution order for the terminal nodes within a stream. You use the stream script dialog box to edit the script that is saved with the current stream.

To access the stream script tab in the Stream Properties dialog box:

- From the Tools menu, choose:
Stream Properties > Execution
- Click the Execution tab to work with scripts for the current stream.

Use the toolbar icons at the top of the stream script dialog box for the following operations:

- Import the contents of a preexisting stand-alone script into the window.
- Save a script as a text file.
- Print a script.
- Append default script.
- Edit a script (undo, cut, copy, paste, and other common edit functions).
- Execute the entire current script.
- Execute selected lines from a script.
- Stop a script during execution. (This icon is only enabled when a script is running.)
- Check the syntax of the script and, if any errors are found, display them for review in the lower pane of the dialog box.

Note: From version 16.0 onwards, SPSS® Modeler uses the Python scripting language. All versions before 16.0 used a scripting language unique to SPSS Modeler, now referred to as Legacy scripting. Depending on the type of script you are working with, on the Execution tab select the Default (optional script) execution mode and then select either Python or Legacy.

You can specify whether a script is or is not run when the stream is executed. To run the script each time the stream is executed, respecting the execution order of the script, select Run this script. This setting provides automation at the stream level for quicker model building. However, the default setting is to ignore this script during stream execution. Even if you select the option Ignore this script, you can always run the script directly from this dialog box.

The script editor includes the following features that help with script authoring:

- Syntax highlighting; keywords, literal values (such as strings and numbers), and comments are highlighted.
- Line numbering.
- Block matching; when the cursor is placed by the start of a program block, the corresponding end block is also highlighted.
- Suggested auto-completion.

The colors and text styles that are used by the syntax highlighter can be customized by using the IBM® SPSS Modeler display preferences. To access the display preferences, choose Tools > Options > User Options and select the Syntax tab.

A list of suggested syntax completions can be accessed by selecting Auto-Suggest from the context menu, or pressing Ctrl + Space. Use the cursor keys to move up and down the list, then press Enter to insert the selected text. To exit from auto-suggest mode without modifying the existing text, press Esc.

The Debug tab displays debugging messages and can be used to evaluate script state once the script is executed. The Debug tab consists of a read-only text area and a single-line input text field. The text area displays text that is sent to either standard output or standard error by the scripts, for example through error message text. The input text field takes input from the user. This input is then evaluated within the context of the script that was most recently executed within the dialog (known as the *scripting context*). The text area contains the command and resulting output so that the user can see a trace of commands. The text input field always contains the command prompt (→ for Legacy scripting).

A new scripting context is created in the following circumstances:

- A script is executed by using either Run this script or Run selected lines.
- The scripting language is changed.

If a new scripting context is created, the text area is cleared.

Note: Executing a stream outside of the script pane does not modify the script context of the script pane. The values of any variables that are created as part of that execution are not visible within the script dialog box.

- [Stream script example: Training a neural net](#)
- [Jython code size limits](#)

Related information

- [Scripting overview](#)
 - [Types of Scripts](#)
 - [Stream script example: Training a neural net](#)
 - [Standalone Scripts](#)
 - [SuperNode Scripts](#)
 - [Standalone script example: Generating a Feature Selection model](#)
-

Stream script example: Training a neural net

A stream can be used to train a neural network model when executed. Normally, to test the model, you might run the modeling node to add the model to the stream, make the appropriate connections, and execute an Analysis node.

Using an IBM® SPSS® Modeler script, you can automate the process of testing the model nugget after you have created it. For example, the following stream script to test the demo stream druglearn.str (available in the /Demos/streams/ folder under your IBM SPSS Modeler installation) could be run from the Stream Properties dialog (Tools > Stream Properties > Script):

```
stream = modeler.script.stream()
neuralnetnode = stream.findByName("neuralnetwork", None)
results = []
neuralnetnode.run(results)
appliernode = stream.createModelApplierAt(results[0], "Drug", 594, 187)
analysisnode = stream.createAt("analysis", "Drug", 688, 187)
typenode = stream.findByName("type", None)
stream.linkBetween(appliernode, typenode, analysisnode)
analysisnode.run([])
```

The following bullets describe each line in this script example.

- The first line defines a variable that points to the current stream.
- In line 2, the script finds the Neural Net builder node.
- In line 3, the script creates a list where the execution results can be stored.
- In line 4, the Neural Net model nugget is created. This is stored in the list defined on line 3.
- In line 5, a model apply node is created for the model nugget and placed on the stream canvas.
- In line 6, an analysis node called **Drug** is created.
- In line 7, the script finds the Type node.
- In line 8, the script connects the model apply node created in line 5 between the Type node and the Analysis node.
- Finally, the Analysis node is executed to produce the Analysis report.

It is possible to use a script to build and run a stream from scratch, starting with a blank canvas.

Jython code size limits

Jython compiles each script to Java bytecode, which is then executed by the Java Virtual Machine (JVM). However, Java imposes a limit on the size of a single bytecode file. So when Jython attempts to load the bytecode, it can cause the JVM to crash. IBM® SPSS® Modeler is unable to prevent this from happening.

Ensure that you write your Jython scripts using good coding practices (such as minimizing duplicated code by using variables or functions to compute common intermediate values). If necessary, you may need to split your code over several source files or define it using modules as these are compiled into bytecode files.

Standalone Scripts

The Standalone Script dialog box is used to create or edit a script that is saved as a text file. It displays the name of the file and provides facilities for loading, saving, importing, and executing scripts.

To access the standalone script dialog box:

From the main menu, choose:

Tools > Standalone Script

The same toolbar and script syntax-checking options are available for standalone scripts as for stream scripts. See the topic [Stream Scripts](#) for more information.

- [Standalone script example: Saving and loading a model](#)
- [Standalone script example: Generating a Feature Selection model](#)

Related information

- [Scripting overview](#)
- [Types of Scripts](#)
- [Stream Scripts](#)
- [Stream script example: Training a neural net](#)
- [SuperNode Scripts](#)
- [Script checking](#)
- [Standalone script example: Saving and loading a model](#)

Standalone script example: Saving and loading a model

Standalone scripts are useful for stream manipulation. Suppose that you have two streams—one that creates a model and another that uses graphs to explore the generated rule set from the first stream with existing data fields. A standalone script for this scenario might look something like this:

```
taskrunner = modeler.script.session().getTaskRunner()

# Modify this to the correct Modeler installation Demos folder.
# Note use of forward slash and trailing slash.
installation = "C:/Program Files/IBM/SPSS/Modeler/19/Demos/"

# First load the model builder stream from file and build a model
druglearn_stream = taskrunner.openStreamFromFile(installation + "streams/druglearn.str", True)
results = []
druglearn_stream.findByType("c50", None).run(results)

# Save the model to file
taskrunner.saveModelToFile(results[0], "rule.gm")

# Now load the plot stream, read the model from file and insert it into the stream
drugplot_stream = taskrunner.openStreamFromFile(installation + "streams/drugplot.str", True)
model = taskrunner.openModelFromFile("rule.gm", True)
modelapplier = drugplot_stream.createModelApplier(model, "Drug")

# Now find the plot node, disconnect it and connect the
# model applier node between the derive node and the plot node
derivenode = drugplot_stream.findByType("derive", None)
plotnode = drugplot_stream.findByType("plot", None)
drugplot_stream.disconnect(plotnode)
modelapplier.setPositionBetween(derivenode, plotnode)
drugplot_stream.linkBetween(modelapplier, derivenode, plotnode)
plotnode.setPropertyValue("color_field", "$C-Drug")
plotnode.run([])
```

Note: To learn more about scripting language in general, see [Scripting language overview](#).

Standalone script example: Generating a Feature Selection model

Starting with a blank canvas, this example builds a stream that generates a Feature Selection model, applies the model, and creates a table that lists the 15 most important fields relative to the specified target.

```
stream = modeler.script.session().createProcessorStream("featureselection", True)

statisticsimportnode = stream.createAt("statisticsimport", "Statistics File", 150, 97)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/customer_dbase.sav")

typenode = stream.createAt("type", "Type", 258, 97)
typenode.setKeyedPropertyValue("direction", "response_01", "Target")

featureselectionnode = stream.createAt("featureselection", "Feature Selection", 366, 97)
featureselectionnode.setPropertyValue("top_n", 15)
featureselectionnode.setPropertyValue("max_missing_values", 80.0)
featureselectionnode.setPropertyValue("selection_mode", "TopN")
featureselectionnode.setPropertyValue("important_label", "Check Me Out!")
featureselectionnode.setPropertyValue("criteria", "Likelihood")

stream.link(statisticsimportnode, typenode)
stream.link(typenode, featureselectionnode)
models = []
featureselectionnode.run(models)

# Assumes the stream automatically places model apply nodes in the stream
applynode = stream.findByType("applyfeatureselection", None)
tablenode = stream.createAt("table", "Table", applynode.getXPosition() + 96, applynode.getYPosition())
stream.link(applynode, tablenode)
tablenode.run([])
```

The script creates a source node to read in the data, uses a Type node to set the role (direction) for the `response_01` field to `Target`, and then creates and executes a Feature Selection node. The script also connects the nodes and positions each on the stream canvas to produce a readable layout. The resulting model nugget is then connected to a Table node, which lists the 15 most important fields as determined by the `selection_mode` and `top_n` properties. See the topic [featureselectionnode.properties](#) for more information.

Related information

- [Stream Scripts](#)
-

SuperNode Scripts

You can create and save scripts within any terminal SuperNodes using the IBM® SPSS® Modeler scripting language. These scripts are only available for terminal SuperNodes and are often used when creating template streams or to impose a special execution order for the SuperNode contents. SuperNode scripts also enable you to have more than one script running within a stream.

For example, let's say you needed to specify the order of execution for a complex stream, and your SuperNode contains several nodes including a SetGlobals node, which needs to be executed before deriving a new field used in a Plot node. In this case, you can create a SuperNode script that executes the SetGlobals node first. Values calculated by this node, such as the average or standard deviation, can then be used when the Plot node is executed.

Within a SuperNode script, you can specify node properties in the same manner as other scripts. Alternatively, you can change and define the properties for any SuperNode or its encapsulated nodes directly from a stream script. See the topic [SuperNode properties](#) for more information. This method works for source and process SuperNodes as well as terminal SuperNodes.

Note: Since only terminal SuperNodes can execute their own scripts, the Scripts tab of the SuperNode dialog box is available only for terminal SuperNodes.

To open the SuperNode script dialog box from the main canvas:

Select a terminal SuperNode on the stream canvas and, from the SuperNode menu, choose:

SuperNode Script...

To open the SuperNode script dialog box from the zoomed-in SuperNode canvas:

Right-click the SuperNode canvas, and from the context menu, choose:

SuperNode Script...

- [SuperNode Script Example](#)

Related information

- [Scripting overview](#)
 - [Types of Scripts](#)
 - [Stream Scripts](#)
 - [Stream script example: Training a neural net](#)
 - [Standalone Scripts](#)
 - [Script checking](#)
 - [SuperNode Script Example](#)
-

SuperNode Script Example

The following SuperNode script declares the order in which the terminal nodes inside the SuperNode should be executed. This order ensures that the Set Globals node is executed first so that the values calculated by this node can then be used when another node is executed.

```
execute 'Set Globals'  
execute 'gains'  
execute 'profit'  
execute 'age v. $CC-pep'  
execute 'Table'
```

Locking and unlocking SuperNodes

The following example illustrates how you can lock and unlock a SuperNode:

```
stream = modeler.script.stream()  
superNode=stream.findByID('id854RNTSD5MB')
```

```

# unlock one super node
print 'unlock the super node with password abcd'
if superNode.unlock('abcd'):
    print 'unlocked.'
else:
    print 'invalid password.'
# lock one super node
print 'lock the super node with password abcd'
superNode.lock('abcd')

```

Related information

- [SuperNode Scripts](#)

Looping and conditional execution in streams

From version 16.0 onwards, SPSS® Modeler enables you to create some basic scripts from within a stream by selecting values within various dialog boxes instead of having to write instructions directly in the scripting language. The two main types of scripts you can create in this way are simple loops and a way to execute nodes if a condition has been met.

You can combine both looping and conditional execution rules within a stream. For example, you may have data relating to sales of cars from manufacturers worldwide. You could set up a loop to process the data in a stream, identifying details by the country of manufacture, and output the data to different graphs showing details such as sales volume by model, emissions levels by both manufacturer and engine size, and so on. If you were interested in analyzing European information only, you could also add conditions to the looping that prevented graphs being created for manufacturers based in America and Asia.

Note: Because both looping and conditional execution are based on background scripts they are only applied to a whole stream when it is run.

- **Looping** You can use looping to automate repetitive tasks. For example, this might mean adding a given number of nodes to a stream and changing one node parameter each time. Alternatively, you could control the running of a stream or branch again and again for a given number of times, as in the following examples:
 - Run the stream a given number of times and change the source each time.
 - Run the stream a given number of times, changing the value of a variable each time.
 - Run the stream a given number of times, entering one extra field on each execution.
 - Build a model a given number of times and change a model setting each time.
- **Conditional Execution** You can use this to control how terminal nodes are run, based on conditions that you predefine, examples may include the following:
 - Based on whether a given value is true or false, control if a node will be run.
 - Define whether looping of nodes will be run in parallel or sequentially.

Both looping and conditional execution are set up on the Execution tab within the Stream Properties dialog box. Any nodes that are used in conditional or looping requirements are shown with an additional symbol attached to them on the stream canvas to indicate that they are taking part in looping and conditional execution.

You can access the Execution tab in one of 3 ways:

- Using the menus at the top of the main dialog box:
 1. From the Tools menu, choose:
[Stream Properties](#) > [Execution](#)
 2. Click the Execution tab to work with scripts for the current stream.
- From within a stream:
 1. Right-click on a node and choose Looping/Conditional Execution.
 2. Select the relevant submenu option.
- From the graphic toolbar at the top of the main dialog box, click the stream properties icon.

If this is the first time you have set up either looping or conditional execution details, on the Execution tab select the Looping/Conditional Execution execution mode and then select either the Conditional or Looping subtab.

- [Looping in streams](#)
- [Conditional execution in streams](#)

Related information

- [Scripting overview](#)
- [Looping in streams](#)
- [Conditional execution in streams](#)
- [Types of Scripts](#)

- [Standalone Scripts](#)
 - [SuperNode Scripts](#)
-

Looping in streams

With looping you can automate repetitive tasks in streams; examples may include the following:

- Run the stream a given number of times and change the source each time.
- Run the stream a given number of times, changing the value of a variable each time.
- Run the stream a given number of times, entering one extra field on each execution.
- Build a model a given number of times and change a model setting each time.

You set up the conditions to be met on the Looping subtab of the stream Execution tab. To display the subtab, select the Looping/Conditional Execution execution mode.

Any looping requirements that you define will take effect when you run the stream, if the Looping/Conditional Execution execution mode has been set. Optionally, you can generate the script code for your looping requirements and paste it into the script editor by clicking Paste... in the bottom right corner of the Looping subtab; the main Execution tab display changes to show the Default (optional script) execution mode with the script in the top part of the tab. This means that you can define a looping structure using the various looping dialog box options before generating a script that you can customize further in the script editor. Note that when you click Paste... any conditional execution requirements you have defined will also be displayed in the generated script.

Important: The looping variables that you set in a SPSS® Modeler stream may be overridden if you run the stream in a IBM® SPSS Collaboration and Deployment Services job. This is because the IBM SPSS Collaboration and Deployment Services job editor entry overrides the SPSS Modeler entry. For example, if you set a looping variable in the stream to create a different output file name for each loop, the files are correctly named in SPSS Modeler but are overridden by the fixed entry entered on the Result tab of the IBM SPSS Collaboration and Deployment Services Deployment Manager.

To set up a loop

1. Create an iteration key to define the main looping structure to be carried out in a stream. See [Create an iteration key](#) for more information.
 2. Where needed, define one or more iteration variables. See [Create an iteration variable](#) for more information.
 3. The iterations and any variables you created are shown in the main body of the subtab. By default, iterations are executed in the order they appear; to move an iteration up or down the list, click on it to select it then use the up or down arrow in the right hand column of the subtab to change the order.
- [Creating an iteration key for looping in streams](#)
 - [Creating an iteration variable for looping in streams](#)
 - [Selecting fields for iterations](#)

Related information

- [Scripting overview](#)
 - [Looping and conditional execution in streams](#)
 - [Conditional execution in streams](#)
 - [Types of Scripts](#)
 - [Standalone Scripts](#)
 - [SuperNode Scripts](#)
-

Creating an iteration key for looping in streams

You use an iteration key to define the main looping structure to be carried out in a stream. For example, if you are analyzing car sales, you could create a stream parameter *Country of manufacture* and use this as the iteration key; when the stream is run this key is set to each different country value in your data during each iteration. Use the Define Iteration Key dialog box to set up the key.

To open the dialog box, either select the Iteration Key... button in the bottom left corner of the Looping subtab, or right click on any node in the stream and select either Looping/Conditional Execution > Define Iteration Key (Fields) or Looping/Conditional Execution > Define Iteration Key (Values). If you open the dialog box from the stream, some of the fields may be completed automatically for you, such as the name of the node.

To set up an iteration key, complete the following fields:

Iterate on. You can select from one of the following options:

- Stream Parameter - Fields. Use this option to create a loop that sets the value of an existing stream parameter to each specified field in turn.
- Stream Parameter - Values. Use this option to create a loop that sets the value of an existing stream parameter to each specified value in turn.
- Node Property - Fields. Use this option to create a loop that sets the value of a node property to each specified field in turn.
- Node Property - Values. Use this option to create a loop that sets the value of a node property to each specified value in turn.

What to Set. Choose the item that will have its value set each time the loop is executed. You can select from one of the following options:

- Parameter. Only available if you select either Stream Parameter - Fields or Stream Parameter - Values. Select the required parameter from the available list.
- Node. Only available if you select either Node Property - Fields or Node Property - Values. Select the node for which you want to set up a loop. Click the browse button to open the Select Node dialog and choose the node you want; if there are too many nodes listed you can filter the display to only show certain types of nodes by selecting one of the following categories: Source, Process, Graph, Modeling, Output, Export, or Apply Model nodes.
- Property. Only available if you select either Node Property - Fields or Node Property - Values. Select the property of the node from the available list.

Fields to Use. Only available if you select either Stream Parameter - Fields or Node Property - Fields. Choose the field, or fields, within a node to use to provide the iteration values. You can select from one of the following options:

- Node. Only available if you select Stream Parameter - Fields. Select the node that contains the details for which you want to set up a loop. Click the browse button to open the Select Node dialog and choose the node you want; if there are too many nodes listed you can filter the display to only show certain types of nodes by selecting one of the following categories: Source, Process, Graph, Modeling, Output, Export, or Apply Model nodes.
- Field List. Click the list button in the right column to display the Select Fields dialog box, within which you select the fields in the node to provide the iteration data. See [Selecting fields for iterations](#) for more information.

Values to Use. Only available if you select either Stream Parameter - Values or Node Property - Values. Choose the value, or values, within the selected field to use as iteration values. You can select from one of the following options:

- Node. Only available if you select Stream Parameter - Values. Select the node that contains the details for which you want to set up a loop. Click the browse button to open the Select Node dialog and choose the node you want; if there are too many nodes listed you can filter the display to only show certain types of nodes by selecting one of the following categories: Source, Process, Graph, Modeling, Output, Export, or Apply Model nodes.
- Field List. Select the field in the node to provide the iteration data.
- Value List. Click the list button in the right column to display the Select Values dialog box, within which you select the values in the field to provide the iteration data.

Related information

- [Scripting overview](#)
 - [Looping and conditional execution in streams](#)
 - [Looping in streams](#)
 - [Conditional execution in streams](#)
 - [Types of Scripts](#)
 - [Standalone Scripts](#)
 - [SuperNode Scripts](#)
-

Creating an iteration variable for looping in streams

You can use iteration variables to change the values of stream parameters or properties of selected nodes within a stream each time a loop is executed. For example, if your stream loop is analyzing car sales data and using *Country of manufacture* as the iteration key, you may have one graph output showing sales by model and another graph output showing exhaust emissions information. In these cases you could create iteration variables that create new titles for the resultant graphs, such as *Swedish vehicle emissions* and *Japanese car sales by model*. Use the Define Iteration Variable dialog box to set up any variables that you require.

To open the dialog box, either select the Add Variable... button in the bottom left corner of the Looping subtab, or right click on any node in the stream and select:Looping/Conditional Execution > Define Iteration Variable.

To set up an iteration variable, complete the following fields:

Change. Select the type of attribute that you want to amend. You can choose from either Stream Parameter or Node Property.

- If you select Stream Parameter, choose the required parameter and then, by using one of the following options, if available in your stream, define what the value of that parameter should be set to with each iteration of the loop:
 - Global variable. Select the global variable that the stream parameter should be set to.

- Table output cell. To set a stream parameter to be the value in a table output cell, select the table from the list and enter the Row and Column to be used.
 - Enter manually. Select this if you want to manually enter a value for this parameter to take in each iteration. When you return to the Looping subtab a new column is created into which you enter the required text.
 - If you select Node Property, choose the required node and one of its properties and then set the value you want to use for that property. Set the new property value by using one of the following options:
 - Alone. The property value will use the iteration key value. See [Creating an iteration key for looping in streams](#) for more information.
 - As prefix to stem. Uses the iteration key value as a prefix to what you enter in the Stem field.
 - As suffix to stem. Uses the iteration key value as a suffix to what you enter in the Stem field
- If you select either the prefix or suffix option you are prompted to add the additional text to the Stem field. For example, if your iteration key value is *Country of manufacture*, and you select As prefix to stem, you might enter - *sales by model* in this field.

Related information

- [Scripting overview](#)
- [Looping and conditional execution in streams](#)
- [Looping in streams](#)
- [Conditional execution in streams](#)
- [Types of Scripts](#)
- [Standalone Scripts](#)
- [SuperNode Scripts](#)

Selecting fields for iterations

When creating iterations you can select one or more fields using the Select Fields dialog box.

Sort by You can sort available fields for viewing by selecting one of the following options:

- Natural View the order of fields as they have been passed down the data stream into the current node.
- Name Use alphabetical order to sort fields for viewing.
- Type View fields sorted by their measurement level. This option is useful when selecting fields with a particular measurement level.

Select fields from the list one at a time or use the Shift-click and Ctrl-click methods to select multiple fields. You can also use the buttons below the list to select groups of fields based on their measurement level, or to select or deselect all fields in the table.

Note that the fields available for selection are filtered to show only the fields that are appropriate for the stream parameter or node property you are using. For example, if you are using a stream parameter that has a storage type of String, only fields that have a storage type of String are shown.

Related information

- [Scripting overview](#)
- [Looping and conditional execution in streams](#)
- [Looping in streams](#)
- [Conditional execution in streams](#)

Conditional execution in streams

With conditional execution you can control how terminal nodes are run, based on the stream contents matching conditions that you define; examples may include the following:

- Based on whether a given value is true or false, control if a node will be run.
- Define whether looping of nodes will be run in parallel or sequentially.

You set up the conditions to be met on the Conditional subtab of the stream Execution tab. To display the subtab, select the Looping/Conditional Execution execution mode.

Any conditional execution requirements that you define will take effect when you run the stream, if the Looping/Conditional Execution execution mode has been set. Optionally, you can generate the script code for your conditional execution requirements and paste it into the script editor by clicking Paste... in the bottom right corner of the Conditional subtab; the main Execution tab display changes to show the Default (optional script) execution mode with the script in the top part of the tab. This means that you can define conditions using the various looping dialog box options before generating a script that you can customize further in the script editor. Note that when you click Paste... any looping requirements you have defined will also be displayed in the generated script.

To set up a condition:



1. In the right hand column of the Conditional subtab, click the Add New Condition button to open the Add Conditional Execution Statement dialog box. In this dialog you specify the condition that must be met in order for the node to be executed.
2. In the Add Conditional Execution Statement dialog box, specify the following:
 - a. **Node**. Select the node for which you want to set up conditional execution. Click the browse button to open the Select Node dialog and choose the node you want; if there are too many nodes listed you can filter the display to show nodes by one of the following categories: Export, Graph, Modeling, or Output node.
 - b. **Condition based on**. Specify the condition that must be met for the node to be executed. You can choose from one of four options: Stream parameter, Global variable, Table output cell, or Always true. The details you enter in the bottom half of the dialog box are controlled by the condition you choose.
 - **Stream parameter**. Select the parameter from the list available and then choose the Operator for that parameter; for example, the operator may be More than, Equals, Less than, Between, and so on. You then enter the Value, or minimum and maximum values, depending on the operator.
 - **Global variable**. Select the variable from the list available; for example, this might include: Mean, Sum, Minimum value, Maximum value, or Standard deviation. You then select the Operator and values required.
 - **Table output cell**. Select the table node from the list available and then choose the Row and Column in the table. You then select the Operator and values required.
 - **Always true**. Select this option if the node must always be executed. If you select this option, there are no further parameters to select.
3. Repeat steps 1 and 2 as often as required until you have set up all the conditions you require. The node you selected and the condition to be met before that node is executed are shown in the main body of the subtab in the Execute Node and If this condition is true columns respectively.
4. By default, nodes and conditions are executed in the order they appear; to move a node and condition up or down the list, click on it to select it then use the up or down arrow in the right hand column of the subtab to change the order.

In addition, you can set the following options at the bottom of the Conditional subtab:

- **Evaluate all in order**. Select this option to evaluate each condition in the order in which they are shown on the subtab. The nodes for which conditions have been found to be "True" will all be executed once all the conditions have been evaluated.
- **Execute one at a time**. Only available if **Evaluate all in order** is selected. Selecting this means that if a condition is evaluated as "True", the node associated with that condition is executed before the next condition is evaluated.
- **Evaluate until first hit**. Selecting this means that only the first node that returns a "True" evaluation from the conditions you specified will be run.

Related information

- [Scripting overview](#)
 - [Looping and conditional execution in streams](#)
 - [Looping in streams](#)
 - [Types of Scripts](#)
 - [Standalone Scripts](#)
 - [SuperNode Scripts](#)
-

Executing and interrupting scripts

A number of ways of executing scripts are available. For example, on the stream script or standalone script dialog, the "Run this script" button executes the complete script:

Figure 1. Run This Script button



The "Run selected lines" button executes a single line, or a block of adjacent lines, that you have selected in the script:

Figure 2. Run Selected Lines button



You can execute a script using any of the following methods:

- Click the "Run this script" or "Run selected lines" button within a stream script or standalone script dialog box.
- Run a stream where Run this script is set as the default execution method.
- Use the `-execute` flag on startup in interactive mode. See the topic [Using command line arguments](#) for more information.

Note: A SuperNode script is executed when the SuperNode is executed as long as you have selected Run this script within the SuperNode script dialog box.

Interrupting script execution

Within the stream script dialog box, the red stop button is activated during script execution. Using this button, you can abandon the execution of the script and any current stream.

The Scripting Language

- [Scripting language overview](#)
 - [Python and Jython](#)
 - [Python Scripting](#)
 - [Object-Oriented Programming](#)
-

Scripting language overview

The scripting facility for IBM® SPSS® Modeler enables you to create scripts that operate on the SPSS Modeler user interface, manipulate output objects, and run command syntax. You can run scripts directly from within SPSS Modeler.

Scripts in IBM SPSS Modeler are written in the scripting language Python. The Java-based implementation of Python that is used by IBM SPSS Modeler is called Jython. The scripting language consists of the following features:

- A format for referencing nodes, streams, projects, output, and other IBM SPSS Modeler objects.
- A set of scripting statements or commands that can be used to manipulate these objects.
- A scripting expression language for setting the values of variables, parameters, and other objects.
- Support for comments, continuations, and blocks of literal text.

The following sections describe the Python scripting language, the Jython implementation of Python, and the basic syntax for getting started with scripting within IBM SPSS Modeler. Information about specific properties and commands is provided in the sections that follow.

Related information

- [Python and Jython](#)
 - [Python Scripting](#)
 - [Object-Oriented Programming](#)
-

Python and Jython

Jython is an implementation of the Python scripting language, which is written in the Java language and integrated with the Java platform. Python is a powerful object-oriented scripting language. Jython is useful because it provides the productivity features of a mature scripting language and, unlike Python, runs in any environment that supports a Java virtual machine (JVM). This means that the Java libraries on the JVM are available to use when you are writing programs. With Jython, you can take advantage of this difference, and use the syntax and most of the features of the Python language.

As a scripting language, Python (and its Jython implementation) is easy to learn and efficient to code, and has minimal required structure to create a running program. Code can be entered interactively, that is, one line at a time. Python is an interpreted scripting language; there is no precompile step, as there is in Java. Python programs are simply text files that are interpreted as they are input (after parsing for syntax errors). Simple expressions, like defined values, as well as more complex actions, such as function definitions, are immediately executed and available for use. Any changes that are made to the code can be tested quickly. Script interpretation does, however, have some disadvantages. For example, use of an undefined variable is not a compiler error, so it is detected only if (and when) the statement in which the variable is used is executed. In this case, the program can be edited and run to debug the error.

Python sees everything, including all data and code, as an object. You can, therefore, manipulate these objects with lines of code. Some select types, such as numbers and strings, are more conveniently considered as values, not objects; this is supported by Python. There is one null value that is supported. This null value has the reserved name **None**.

For a more in-depth introduction to Python and Jython scripting, and for some example scripts, see <http://www.ibm.com/developerworks/java/tutorials/j-jython1/j-jython1.html> and <http://www.ibm.com/developerworks/java/tutorials/j-jython2/j-jython2.html>.

Related information

- [Scripting language overview](#)
 - [Python Scripting](#)
 - [Object-Oriented Programming](#)
-

Python Scripting

This guide to the Python scripting language is an introduction to the components that are most likely to be used when scripting in IBM® SPSS® Modeler, including concepts and programming basics. This will provide you with enough knowledge to start developing your own Python scripts to use within IBM SPSS Modeler.

- [Operations](#)
- [Lists](#)
- [Strings](#)
- [Remarks](#)
- [Statement Syntax](#)
- [Identifiers](#)
- [Blocks of Code](#)
- [Passing Arguments to a Script](#)
- [Examples](#)
- [Mathematical Methods](#)
- [Using Non-ASCII characters](#)

Related information

- [Operations](#)
 - [Lists](#)
 - [Strings](#)
 - [Remarks](#)
 - [Statement Syntax](#)
 - [Identifiers](#)
 - [Blocks of Code](#)
 - [Passing Arguments to a Script](#)
 - [Examples](#)
 - [Mathematical Methods](#)
 - [Using Non-ASCII characters](#)
 - [Scripting language overview](#)
 - [Python and Jython](#)
 - [Object-Oriented Programming](#)
-

Operations

Assignment is done using an equals sign (=). For example, to assign the value "3" to a variable called "x" you would use the following statement:

```
x = 3
```

The equals sign is also used to assign string type data to a variable. For example, to assign the value "a string value" to the variable "y" you would use the following statement:

```
y = "a string value"
```

The following table lists some commonly used comparison and numeric operations, and their descriptions.

Table 1. Common comparison and numeric operations

Operation	Description
<code>x < y</code>	Is <code>x</code> less than <code>y</code> ?
<code>x > y</code>	Is <code>x</code> greater than <code>y</code> ?
<code>x <= y</code>	Is <code>x</code> less than or equal to <code>y</code> ?
<code>x >= y</code>	Is <code>x</code> greater than or equal to <code>y</code> ?
<code>x == y</code>	Is <code>x</code> equal to <code>y</code> ?
<code>x != y</code>	Is <code>x</code> not equal to <code>y</code> ?
<code>x <> y</code>	Is <code>x</code> not equal to <code>y</code> ?

Operation	Description
<code>x + y</code>	Add <code>y</code> to <code>x</code>
<code>x - y</code>	Subtract <code>y</code> from <code>x</code>
<code>x * y</code>	Multiply <code>x</code> by <code>y</code>
<code>x / y</code>	Divide <code>x</code> by <code>y</code>
<code>x ** y</code>	Raise <code>x</code> to the <code>y</code> power

Related information

- [Python Scripting](#)
 - [Lists](#)
 - [Strings](#)
 - [Remarks](#)
 - [Statement Syntax](#)
 - [Identifiers](#)
 - [Blocks of Code](#)
 - [Passing Arguments to a Script](#)
 - [Examples](#)
 - [Mathematical Methods](#)
-

Lists

Lists are sequences of elements. A list can contain any number of elements, and the elements of the list can be any type of object. Lists can also be thought of as arrays. The number of elements in a list can increase or decrease as elements are added, removed, or replaced.

Examples

<code>[]</code>	Any empty list.
<code>[1]</code>	A list with a single element, an integer.
<code>["Mike", 10, "Don", 20]</code>	A list with four elements, two string elements and two integer elements.
<code>[[], [7], [8, 9]]</code>	A list of lists. Each sub-list is either an empty list or a list of integer elements.
<code>x = 7; y = 2; z = 3; [1, x, y, x + y]</code>	A list of integers. This example demonstrates the use of variables and expressions.

You can assign a list to a variable, for example:

```
mylist1 = ["one", "two", "three"]
```

You can then access specific elements of the list, for example:

```
mylist[0]
```

This will result in the following output:

```
one
```

The number in the brackets (`[]`) is known as an *index* and refers to a particular element of the list. The elements of a list are indexed starting from 0.

You can also select a range of elements of a list; this is called *slicing*. For example, `x[1:3]` selects the second and third elements of `x`. The end index is one past the selection.

Related information

- [Python Scripting](#)
- [Operations](#)
- [Strings](#)
- [Remarks](#)
- [Statement Syntax](#)
- [Identifiers](#)
- [Blocks of Code](#)
- [Passing Arguments to a Script](#)
- [Examples](#)
- [Mathematical Methods](#)

Strings

A *string* is an immutable sequence of characters that is treated as a value. Strings support all of the immutable sequence functions and operators that result in a new string. For example, "`abcdef[1:4]`" results in the output "`bcd`".

In Python, characters are represented by strings of length one.

String literals are defined by the use of single or triple quoting. Strings that are defined using single quotes cannot span lines, while strings that are defined using triple quotes can. A string can be enclosed in single quotes ('') or double quotes (""). A quoting character may contain the other quoting character un-escaped or the quoting character escaped, that is preceded by the backslash (\) character.

Examples

```
"This is a string"
'This is also a string'
"It's a string"
'This book is called "Python Scripting and Automation Guide".'
"This is an escape quote (\") in a quoted string"
```

Multiple strings separated by white space are automatically concatenated by the Python parser. This makes it easier to enter long strings and to mix quote types in a single string, for example:

```
"This string uses ' and " that string uses ". '
```

This results in the following output:

```
This string uses ' and that string uses ".
```

Strings support several useful methods. Some of these methods are given in the following table.

Table 1. String methods

Method	Usage
<code>s.capitalize()</code>	Initial capitalize <code>s</code>
<code>s.count(ss [,start [,end]])</code>	Count the occurrences of <code>ss</code> in <code>s[start:end]</code>
<code>s.startswith(str [, start [, end]])</code>	Test to see if <code>s</code> starts with <code>str</code>
<code>s.endswith(str [, start [, end]])</code>	Test to see if <code>s</code> ends with <code>str</code>
<code>s.expandtabs({size})</code>	Replace tabs with spaces, default <code>size</code> is 8
<code>s.find(str [, start [, end]])</code>	Finds first index of <code>str</code> in <code>s</code> ; if not found, the result is -1. <code>rfind</code> searches right to left.
<code>s.index(str [, start [, end]])</code>	Finds first index of <code>str</code> in <code>s</code> ; if not found: raise <code>ValueError</code> . <code>rindex</code> searches right to left.
<code>s.isalnum</code>	Test to see if the string is alphanumeric
<code>s.isalpha</code>	Test to see if the string is alphabetic
<code>s.isnum</code>	Test to see if the string is numeric
<code>s.isupper</code>	Test to see if the string is all uppercase
<code>s.islower</code>	Test to see if the string is all lowercase
<code>s.isspace</code>	Test to see if the string is all whitespace
<code>s.istitle</code>	Test to see if the string is a sequence of initial cap alphanumeric strings
<code>s.lower()</code>	Convert to all lower case
<code>s.upper()</code>	Convert to all upper case
<code>s.swapcase()</code>	Convert to all opposite case
<code>s.title()</code>	Convert to all title case
<code>s.join(seq)</code>	Join the strings in <code>seq</code> with <code>s</code> as the separator
<code>s.splitlines({keep})</code>	Split <code>s</code> into lines, if <code>keep</code> is <code>true</code> , keep the new lines
<code>s.split({sep [, max]})</code>	Split <code>s</code> into "words" using <code>sep</code> (default <code>sep</code> is a white space) for up to <code>max</code> times
<code>s.ljust(width)</code>	Left justify the string in a field <code>width</code> wide
<code>s.rjust(width)</code>	Right justify the string in a field <code>width</code> wide
<code>s.center(width)</code>	center justify the string in a field <code>width</code> wide
<code>s.zfill(width)</code>	Fill with 0.
<code>s.lstrip()</code>	Remove leading white space
<code>s.rstrip()</code>	Remove trailing white space
<code>s.strip()</code>	Remove leading and trailing white space
<code>s.translate(str {,delc})</code>	Translate <code>s</code> using table, after removing any characters in <code>delc</code> . <code>str</code> should be a string with length == 256.
<code>s.replace(old, new [, max])</code>	Replaces all or <code>max</code> occurrences of string <code>old</code> with string <code>new</code>

Related information

- [Python Scripting](#)
 - [Operations](#)
 - [Lists](#)
 - [Remarks](#)
 - [Statement Syntax](#)
 - [Identifiers](#)
 - [Blocks of Code](#)
 - [Passing Arguments to a Script](#)
 - [Examples](#)
 - [Mathematical Methods](#)
-

Remarks

Remarks are comments that are introduced by the pound (or hash) sign (#). All text that follows the pound sign on the same line is considered part of the remark and is ignored. A remark can start in any column. The following example demonstrates the use of remarks:

```
#The HelloWorld application is one of the most simple
print 'Hello World' # print the Hello World line
```

Related information

- [Python Scripting](#)
 - [Operations](#)
 - [Lists](#)
 - [Strings](#)
 - [Statement Syntax](#)
 - [Identifiers](#)
 - [Blocks of Code](#)
 - [Passing Arguments to a Script](#)
 - [Examples](#)
 - [Mathematical Methods](#)
-

Statement Syntax

The statement syntax for Python is very simple. In general, each source line is a single statement. Except for **expression** and **assignment** statements, each statement is introduced by a keyword name, such as **if** or **for**. Blank lines or remark lines can be inserted anywhere between any statements in the code. If there is more than one statement on a line, each statement must be separated by a semicolon (;).

Very long statements can continue on more than one line. In this case the statement that is to continue on to the next line must end with a backslash (\), for example:

```
x = "A loooooooooooooong string" + \
    "another loooooooooooooong string"
```

When a structure is enclosed by parentheses (()), brackets ([]), or curly braces ({ }), the statement can be continued on to a new line after any comma, without having to insert a backslash, for example:

```
x = (1, 2, 3, "hello",
      "goodbye", 4, 5, 6)
```

Related information

- [Python Scripting](#)
- [Operations](#)
- [Lists](#)
- [Strings](#)
- [Remarks](#)
- [Identifiers](#)
- [Blocks of Code](#)
- [Passing Arguments to a Script](#)
- [Examples](#)

- [Mathematical Methods](#)
-

Identifiers

Identifiers are used to name variables, functions, classes and keywords. Identifiers can be any length, but must start with either an alphabetical character of upper or lower case, or the underscore character (_). Names that start with an underscore are generally reserved for internal or private names. After the first character, the identifier can contain any number and combination of alphabetical characters, numbers from 0-9, and the underscore character.

There are some reserved words in Jython that cannot be used to name variables, functions, or classes. They fall under the following categories:

- **Statement introducers:** assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, pass, print, raise, return, try, and while
- **Parameter introducers:** as, import, and in
- **Operators:** and, in, is, lambda, not, and or

Improper keyword use generally results in a SyntaxError.

Related information

- [Python Scripting](#)
 - [Operations](#)
 - [Lists](#)
 - [Strings](#)
 - [Remarks](#)
 - [Statement Syntax](#)
 - [Blocks of Code](#)
 - [Passing Arguments to a Script](#)
 - [Examples](#)
 - [Mathematical Methods](#)
-

Blocks of Code

Blocks of code are groups of statements that are used where single statements are expected. Blocks of code can follow any of the following statements: if, elif, else, for, while, try, except, def, and class. These statements introduce the block of code with the colon character (:), for example:

```
if x == 1:  
    y = 2  
    z = 3  
elif:  
    y = 4  
    z = 5
```

Indentation is used to delimit code blocks (rather than the curly braces that are used in Java). All lines in a block must be indented to the same position. This is because a change in the indentation indicates the end of a code block. It is usual to indent by four spaces per level. It is recommended that spaces are used to indent the lines, rather than tabs. Spaces and tabs must not be mixed. The lines in the outermost block of a module must start at column one, else a SyntaxError will occur.

The statements that make up a code block (and follow the colon) can also be on a single line, separated by semicolons, for example:

```
if x == 1: y = 2; z = 3;
```

Related information

- [Python Scripting](#)
- [Operations](#)
- [Lists](#)
- [Strings](#)
- [Remarks](#)
- [Statement Syntax](#)
- [Identifiers](#)
- [Passing Arguments to a Script](#)
- [Examples](#)
- [Mathematical Methods](#)

Passing Arguments to a Script

Passing arguments to a script is useful as it means a script can be used repeatedly without modification. The arguments that are passed on the command line are passed as values in the list `sys.argv`. The number of values passed can be obtained by using the command `len(sys.argv)`. For example:

```
import sys
print "test1"
print sys.argv[0]
print sys.argv[1]
print len(sys.argv)
```

In this example, the `import` command imports the entire `sys` class so that the methods that exist for this class, such as `argv`, can be used.

The script in this example can be invoked using the following line:

```
/u/mjloos/test1 mike don
```

The result is the following output:

```
/u/mjloos/test1 mike don
test1
mike
don
3
```

Related information

- [Python Scripting](#)
- [Operations](#)
- [Lists](#)
- [Strings](#)
- [Remarks](#)
- [Statement Syntax](#)
- [Identifiers](#)
- [Blocks of Code](#)
- [Examples](#)
- [Mathematical Methods](#)

Examples

The `print` keyword prints the arguments immediately following it. If the statement is followed by a comma, a new line is not included in the output. For example:

```
print "This demonstrates the use of a",
print " comma at the end of a print statement."
```

This will result in the following output:

```
This demonstrates the use of a comma at the end of a print statement.
```

The `for` statement is used to iterate through a block of code. For example:

```
mylist1 = ["one", "two", "three"]
for lv in mylist1:
    print lv
    continue
```

In this example, three strings are assigned to the list `mylist1`. The elements of the list are then printed, with one element of each line. This will result in the following output:

```
one
two
three
```

In this example, the iterator `lv` takes the value of each element in the list `mylist1` in turn as the for loop implements the code block for each element. An iterator can be any valid identifier of any length.

The `if` statement is a conditional statement. It evaluates the condition and returns either true or false, depending on the result of the evaluation. For example:

```

mylist1 = ["one", "two", "three"]
for lv in mylist1:
    if lv == "two":
        print "The value of lv is ", lv
    else
        print "The value of lv is not two, but ", lv
    continue

```

In this example, the value of the iterator `lv` is evaluated. If the value of `lv` is `two` a different string is returned to the string that is returned if the value of `lv` is not `two`. This results in the following output:

```

The value of lv is not two, but one
The value of lv is two
The value of lv is not two, but three

```

Related information

- [Python Scripting](#)
 - [Operations](#)
 - [Lists](#)
 - [Strings](#)
 - [Remarks](#)
 - [Statement Syntax](#)
 - [Identifiers](#)
 - [Blocks of Code](#)
 - [Passing Arguments to a Script](#)
 - [Mathematical Methods](#)
-

Mathematical Methods

From the `math` module you can access useful mathematical methods. Some of these methods are given in the following table. Unless specified otherwise, all values are returned as floats.

Table 1. Mathematical methods

Method	Usage
<code>math.ceil(x)</code>	Return the ceiling of <code>x</code> as a float, that is the smallest integer greater than or equal to <code>x</code>
<code>math.copysign(x, y)</code>	Return <code>x</code> with the sign of <code>y</code> . <code>copysign(1, -0.0)</code> returns <code>-1</code>
<code>math.fabs(x)</code>	Return the absolute value of <code>x</code>
<code>math.factorial(x)</code>	Return <code>x</code> factorial. If <code>x</code> is negative or not an integer, a <code>ValueError</code> is raised.
<code>math.floor(x)</code>	Return the floor of <code>x</code> as a float, that is the largest integer less than or equal to <code>x</code>
<code>math.frexp(x)</code>	Return the mantissa (<code>m</code>) and exponent (<code>e</code>) of <code>x</code> as the pair (<code>m, e</code>). <code>m</code> is a float and <code>e</code> is an integer, such that <code>x == m * 2**e</code> exactly. If <code>x</code> is zero, returns <code>(0.0, 0)</code> , otherwise <code>0.5 <= abs(m) < 1</code> .
<code>math.fsum(iterable)</code>	Return an accurate floating point sum of values in <code>iterable</code>
<code>math.isinf(x)</code>	Check if the float <code>x</code> is positive or negative infinitive
<code>math.isnan(x)</code>	Check if the float <code>x</code> is <code>NaN</code> (not a number)
<code>math.ldexp(x, i)</code>	Return <code>x * (2**i)</code> . This is essentially the inverse of the function <code>frexp</code> .
<code>math.modf(x)</code>	Return the fractional and integer parts of <code>x</code> . Both results carry the sign of <code>x</code> and are floats.
<code>math.trunc(x)</code>	Return the <code>Real</code> value <code>x</code> , that has been truncated to an <code>Integral</code> .
<code>math.exp(x)</code>	Return <code>e**x</code>
<code>math.log(x[, base])</code>	Return the logarithm of <code>x</code> to the given value of <code>base</code> . If <code>base</code> is not specified, the natural logarithm of <code>x</code> is returned.
<code>math.log1p(x)</code>	Return the natural logarithm of <code>1+x</code> (<code>base e</code>)
<code>math.log10(x)</code>	Return the base-10 logarithm of <code>x</code>
<code>math.pow(x, y)</code>	Return <code>x</code> raised to the power <code>y</code> . <code>pow(1.0, x)</code> and <code>pow(x, 0.0)</code> always return 1, even when <code>x</code> is zero or <code>NaN</code> .
<code>math.sqrt(x)</code>	Return the square root of <code>x</code>

In addition to the mathematical functions, there are some useful trigonometric methods. These methods are shown in the following table.

Table 2. Trigonometric methods

Method	Usage
<code>math.acos(x)</code>	Return the arc cosine of <code>x</code> in radians
<code>math.asin(x)</code>	Return the arc sine of <code>x</code> in radians
<code>math.atan(x)</code>	Return the arc tangent of <code>x</code> in radians
<code>math.atan2(y, x)</code>	Return <code>atan(y / x)</code> in radians.
<code>math.cos(x)</code>	Return the cosine of <code>x</code> in radians.
<code>math.hypot(x, y)</code>	Return the Euclidean norm <code>sqrt(x*x + y*y)</code> . This is the length of the vector from the origin to the point <code>(x, y)</code> .
<code>math.sin(x)</code>	Return the sine of <code>x</code> in radians
<code>math.tan(x)</code>	Return the tangent of <code>x</code> in radians
<code>math.degrees(x)</code>	Convert angle <code>x</code> from radians to degrees
<code>math.radians(x)</code>	Convert angle <code>x</code> from degrees to radians
<code>math.acosh(x)</code>	Return the inverse hyperbolic cosine of <code>x</code>
<code>math.asinh(x)</code>	Return the inverse hyperbolic sine of <code>x</code>
<code>math.atanh(x)</code>	Return the inverse hyperbolic tangent of <code>x</code>
<code>math.cosh(x)</code>	Return the hyperbolic cosine of <code>x</code>
<code>math.sinh(x)</code>	Return the hyperbolic sine of <code>x</code>
<code>math.tanh(x)</code>	Return the hyperbolic tangent of <code>x</code>

There are also two mathematical constants. The value of `math.pi` is the mathematical constant pi. The value of `math.e` is the mathematical constant e.

Related information

- [Python Scripting](#)
 - [Operations](#)
 - [Lists](#)
 - [Strings](#)
 - [Remarks](#)
 - [Statement Syntax](#)
 - [Identifiers](#)
 - [Blocks of Code](#)
 - [Passing Arguments to a Script](#)
 - [Examples](#)
-

Using Non-ASCII characters

In order to use non-ASCII characters, Python requires explicit encoding and decoding of strings into Unicode. In IBM® SPSS® Modeler, Python scripts are assumed to be encoded in UTF-8, which is a standard Unicode encoding that supports non-ASCII characters. The following script will compile because the Python compiler has been set to UTF-8 by SPSS Modeler.

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", "テストノード", 96, 64)
```

However, the resulting node will have an incorrect label.

Figure 1. Node label containing non-ASCII characters, displayed incorrectly



äfää,äf^äf äf%äf%

The label is incorrect because the string literal itself has been converted to an ASCII string by Python.

Python allows Unicode string literals to be specified by adding a `u` character prefix before the string literal:

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", u"テストノード", 96, 64)
```

This will create a Unicode string and the label will appear correctly.

Figure 2. Node label containing non-ASCII characters, displayed correctly



テストノード

Using Python and Unicode is a large topic which is beyond the scope of this document. Many books and online resources are available that cover this topic in great detail.

Related information

- [Python Scripting](#)
 - [Operations](#)
 - [Lists](#)
 - [Strings](#)
 - [Remarks](#)
 - [Statement Syntax](#)
 - [Identifiers](#)
 - [Blocks of Code](#)
 - [Passing Arguments to a Script](#)
 - [Examples](#)
-

Object-Oriented Programming

Object-oriented programming is based on the notion of creating a model of the target problem in your programs. Object-oriented programming reduces programming errors and promotes the reuse of code. Python is an object-oriented language. Objects defined in Python have the following features:

- **Identity.** Each object must be distinct, and this must be testable. The `is` and `is not` tests exist for this purpose.
- **State.** Each object must be able to store state. Attributes, such as fields and instance variables, exist for this purpose.
- **Behavior.** Each object must be able to manipulate its state. Methods exist for this purpose.

Python includes the following features for supporting object-oriented programming:

- **Class-based object creation.** Classes are templates for the creation of objects. Objects are data structures with associated behavior.
 - **Inheritance with polymorphism.** Python supports single and multiple inheritance. All Python instance methods are polymorphic and can be overridden by subclasses.
 - **Encapsulation with data hiding.** Python allows attributes to be hidden. When hidden, attributes can be accessed from outside the class only through methods of the class. Classes implement methods to modify the data.
- [Defining a Class](#)
 - [Creating a Class Instance](#)
 - [Adding Attributes to a Class Instance](#)
 - [Defining Class Attributes and Methods](#)
 - [Hidden Variables](#)
 - [Inheritance](#)

Related information

- [Defining a Class](#)
 - [Creating a Class Instance](#)
 - [Adding Attributes to a Class Instance](#)
 - [Defining Class Attributes and Methods](#)
 - [Hidden Variables](#)
 - [Inheritance](#)
 - [Scripting language overview](#)
 - [Python and Jython](#)
 - [Python Scripting](#)
-

Defining a Class

Within a Python class, both variables and methods can be defined. Unlike in Java, in Python you can define any number of public classes per source file (or *module*). Therefore, a module in Python can be thought of similar to a package in Java.

In Python, classes are defined using the `class` statement. The `class` statement has the following form:

```
class name (superclasses): statement
```

or

```
class name (superclasses):  
    assignment  
    .  
    .  
    function  
    .  
    .
```

When you define a class, you have the option to provide zero or more *assignment* statements. These create class attributes that are shared by all instances of the class. You can also provide zero or more *function* definitions. These function definitions create methods. The superclasses list is optional.

The class name should be unique in the same scope, that is within a module, function or class. You can define multiple variables to reference the same class.

Related information

- [Object-Oriented Programming](#)
 - [Creating a Class Instance](#)
 - [Adding Attributes to a Class Instance](#)
 - [Defining Class Attributes and Methods](#)
 - [Hidden Variables](#)
 - [Inheritance](#)
-

Creating a Class Instance

Classes are used to hold class (or shared) attributes or to create class instances. To create an instance of a class, you call the class as if it were a function. For example, consider the following class:

```
class MyClass:  
    pass
```

Here, the `pass` statement is used because a statement is required to complete the class, but no action is required programmatically.

The following statement creates an instance of the class `MyClass`:

```
x = MyClass()
```

Related information

- [Object-Oriented Programming](#)
 - [Defining a Class](#)
 - [Adding Attributes to a Class Instance](#)
 - [Defining Class Attributes and Methods](#)
 - [Hidden Variables](#)
 - [Inheritance](#)
-

Adding Attributes to a Class Instance

Unlike in Java, in Python clients can add attributes to an instance of a class. Only the one instance is changed. For example, to add attributes to an instance `x`, set new values on that instance:

```
x.attr1 = 1  
x.attr2 = 2  
.  
. .  
x.attrN = n
```

Related information

- [Object-Oriented Programming](#)
 - [Defining a Class](#)
 - [Creating a Class Instance](#)
 - [Defining Class Attributes and Methods](#)
 - [Hidden Variables](#)
 - [Inheritance](#)
-

Defining Class Attributes and Methods

Any variable that is bound in a class is a *class attribute*. Any function defined within a class is a *method*. Methods receive an instance of the class, conventionally called `self`, as the first argument. For example, to define some class attributes and methods, you might enter the following code:

```
class MyClass
    attr1 = 10      #class attributes
    attr2 = "hello"

    def method1(self):
        print MyClass.attr1  #reference the class attribute

    def method2(self):
        print MyClass.attr2  #reference the class attribute

    def method3(self, text):
        self.text = text      #instance attribute
        print text, self.text  #print my argument and my attribute

    method4 = method3  #make an alias for method3
```

Inside a class, you should qualify all references to class attributes with the class name; for example, `MyClass.attr1`. All references to instance attributes should be qualified with the `self` variable; for example, `self.text`. Outside the class, you should qualify all references to class attributes with the class name (for example `MyClass.attr1`) or with an instance of the class (for example `x.attr1`, where `x` is an instance of the class). Outside the class, all references to instance variables should be qualified with an instance of the class; for example, `x.text`.

Related information

- [Object-Oriented Programming](#)
 - [Defining a Class](#)
 - [Creating a Class Instance](#)
 - [Adding Attributes to a Class Instance](#)
 - [Hidden Variables](#)
 - [Inheritance](#)
-

Hidden Variables

Data can be hidden by creating *Private* variables. Private variables can be accessed only by the class itself. If you declare names of the form `__xxx` or `__xxx_yyy`, that is with two preceding underscores, the Python parser will automatically add the class name to the declared name, creating hidden variables, for example:

```
class MyClass:
    __attr = 10  #private class attribute

    def method1(self):
        pass

    def method2(self, p1, p2):
        pass

    def __privateMethod(self, text):
        self.__text = text  #private attribute
```

Unlike in Java, in Python all references to instance variables must be qualified with `self`; there is no implied use of `this`.

Related information

- [Object-Oriented Programming](#)
 - [Defining a Class](#)
 - [Creating a Class Instance](#)
 - [Adding Attributes to a Class Instance](#)
 - [Defining Class Attributes and Methods](#)
 - [Inheritance](#)
-

Inheritance

The ability to inherit from classes is fundamental to object-oriented programming. Python supports both single and multiple inheritance. *Single inheritance* means that there can be only one superclass. *Multiple inheritance* means that there can be more than one superclass.

Inheritance is implemented by subclassing other classes. Any number of Python classes can be superclasses. In the Jython implementation of Python, only one Java class can be directly or indirectly inherited from. It is not required for a superclass to be supplied.

Any attribute or method in a superclass is also in any subclass and can be used by the class itself, or by any client as long as the attribute or method is not hidden. Any instance of a subclass can be used wherever and instance of a superclass can be used; this is an example of *polymorphism*. These features enable reuse and ease of extension.

Example

```
class Class1: pass      #no inheritance
class Class2: pass
class Class3(Class1): pass      #single inheritance
class Class4(Class3, Class2): pass      #multiple inheritance
```

Related information

- [Object-Oriented Programming](#)
 - [Defining a Class](#)
 - [Creating a Class Instance](#)
 - [Adding Attributes to a Class Instance](#)
 - [Defining Class Attributes and Methods](#)
 - [Hidden Variables](#)
-

Scripting in IBM® SPSS® Modeler

- [Types of scripts](#)
 - [Streams, SuperNode streams, and diagrams](#)
 - [Executing a stream](#)
 - [The scripting context](#)
 - [Referencing existing nodes](#)
 - [Creating nodes and modifying streams](#)
 - [Clearing, or removing, items](#)
 - [Getting information about nodes](#)
-

Types of scripts

In IBM® SPSS® Modeler there are three types of script:

- *Stream scripts* are used to control execution of a single stream and are stored within the stream.
- *SuperNode scripts* are used to control the behavior of SuperNodes.
- *Stand-alone or session scripts* can be used to coordinate execution across a number of different streams.

Various methods are available to be used in scripts in IBM SPSS Modeler with which you can access a wide range of SPSS Modeler functionality. These methods are also used in [The Scripting API](#) to create more advanced functions.

Related information

- [Executing a stream](#)
- [The scripting context](#)
- [Referencing existing nodes](#)
- [Creating nodes and modifying streams](#)
- [Getting information about nodes](#)

Streams, SuperNode streams, and diagrams

Most of the time, the term *stream* means the same thing, regardless of whether it is a stream that is loaded from a file or used within a SuperNode. It generally means a collection of nodes that are connected together and can be executed. In scripting, however, not all operations are supported in all places, meaning a script author should be aware of which stream variant they are using.

- [Streams](#)
- [SuperNode streams](#)
- [Diagrams](#)

Related information

- [Executing a stream](#)
- [The scripting context](#)
- [Referencing existing nodes](#)
- [Creating nodes and modifying streams](#)
- [Getting information about nodes](#)
- [Streams](#)
- [SuperNode streams](#)
- [Diagrams](#)

Streams

A stream is the main IBM® SPSS® Modeler document type. It can be saved, loaded, edited and executed. Streams can also have parameters, global values, a script, and other information associated with them.

Related information

- [Executing a stream](#)
- [The scripting context](#)
- [Referencing existing nodes](#)
- [Creating nodes and modifying streams](#)
- [Getting information about nodes](#)
- [Streams, SuperNode streams, and diagrams](#)

SuperNode streams

A *SuperNode stream* is the type of stream used within a SuperNode. Like a normal stream, it contains nodes which are linked together. SuperNode streams have a number of differences from a normal stream:

- Parameters and any scripts are associated with the SuperNode that owns the SuperNode stream, rather than with the SuperNode stream itself.
- SuperNode streams have additional input and output connector nodes, depending on the type of SuperNode. These connector nodes are used to flow information into and out of the SuperNode stream, and are created automatically when the SuperNode is created.

Related information

- [Executing a stream](#)
- [The scripting context](#)
- [Referencing existing nodes](#)
- [Creating nodes and modifying streams](#)
- [Getting information about nodes](#)
- [Streams, SuperNode streams, and diagrams](#)

Diagrams

The term *diagram* covers the functions that are supported by both normal streams and SuperNode streams, such as adding and removing nodes, and modifying connections between the nodes.

Related information

- [Executing a stream](#)
- [The scripting context](#)
- [Referencing existing nodes](#)
- [Creating nodes and modifying streams](#)
- [Getting information about nodes](#)
- [Streams, SuperNode streams, and diagrams](#)

Executing a stream

The following example runs all executable nodes in the stream, and is the simplest type of stream script:

```
modeler.script.stream().runAll(None)
```

The following example also runs all executable nodes in the stream:

```
stream = modeler.script.stream()
stream.runAll(None)
```

In this example, the stream is stored in a variable called `stream`. Storing the stream in a variable is useful because a script is typically used to modify either the stream or the nodes within a stream. Creating a variable that stores the stream results in a more concise script.

Related information

- [Types of scripts](#)
- [The scripting context](#)
- [Referencing existing nodes](#)
- [Creating nodes and modifying streams](#)
- [Getting information about nodes](#)

The scripting context

The `modeler.script` module provides the context in which a script is executed. The module is automatically imported into a SPSS® Modeler script at run time. The module defines four functions that provide a script with access to its execution environment:

- The `session()` function returns the session for the script. The session defines information such as the locale and the SPSS Modeler backend (either a local process or a networked SPSS Modeler Server) that is being used to run any streams.
- The `stream()` function can be used with stream and SuperNode scripts. This function returns the stream that owns either the stream script or the SuperNode script that is being run.
- The `diagram()` function can be used with SuperNode scripts. This function returns the diagram within the SuperNode. For other script types, this function returns the same as the `stream()` function.
- The `supernode()` function can be used with SuperNode scripts. This function returns the SuperNode that owns the script that is being run.

The four functions and their outputs are summarized in the following table.

Table 1. Summary of `modeler.script` functions

Script type	<code>session()</code>	<code>stream()</code>	<code>diagram()</code>	<code>supernode()</code>
Standalone	Returns a session	Returns the current managed stream at the time the script was invoked (for example, the stream passed via the batch mode <code>-stream</code> option), or <code>None</code> .	Same as for <code>stream()</code>	Not applicable
Stream	Returns a session	Returns a stream	Same as for <code>stream()</code>	Not applicable

Script type	session()	stream()	diagram()	supernode()
SuperNode	Returns a session	Returns a stream	Returns a SuperNode stream	Returns a SuperNode

The `modeler.script` module also defines a way of terminating the script with an exit code. The `exit(exit-code)` function stops the script from executing and returns the supplied integer exit code.

One of the methods that is defined for a stream is `runAll(List)`. This method runs all executable nodes. Any models or outputs that are generated by executing the nodes are added to the supplied list.

It is common for a stream execution to generate outputs such as models, graphs, and other output. To capture this output, a script can supply a variable that is initialized to a list, for example:

```
stream = modeler.script.stream()
results = []
stream.runAll(results)
```

When execution is complete, any objects that are generated by the execution can be accessed from the `results` list.

Related information

- [Types of scripts](#)
- [Executing a stream](#)
- [Referencing existing nodes](#)
- [Creating nodes and modifying streams](#)
- [Getting information about nodes](#)

Referencing existing nodes

A stream is often pre-built with some parameters that must be modified before the stream is executed. Modifying these parameters involves the following tasks:

1. Locating the nodes in the relevant stream.
 2. Changing the node or stream settings (or both).
- [Finding nodes](#)
 - [Setting properties](#)

Related information

- [Finding nodes](#)
- [Setting properties](#)
- [Types of scripts](#)
- [Executing a stream](#)
- [The scripting context](#)
- [Creating nodes and modifying streams](#)
- [Getting information about nodes](#)

Finding nodes

Streams provide a number of ways of locating an existing node. These methods are summarized in the following table.

Table 1. Methods for locating an existing node

Method	Return type	Description
<code>s.findAll(type, label)</code>	Collection	Returns a list of all nodes with the specified type and label. Either the type or label can be <code>None</code> , in which case the other parameter is used.
<code>s.findAll(filter, recursive)</code>	Collection	Returns a collection of all nodes that are accepted by the specified filter. If the recursive flag is <code>True</code> , any SuperNodes within the specified stream are also searched.
<code>s.findById(id)</code>	Node	Returns the node with the supplied ID or <code>None</code> if no such node exists. The search is limited to the current stream.

Method	Return type	Description
<code>s.findByType(type, label)</code>	Node	Returns the node with the supplied type, label, or both. Either the type or name can be <code>None</code> , in which case the other parameter is used. If multiple nodes result in a match, then an arbitrary one is chosen and returned. If no nodes result in a match, then the return value is <code>None</code> .
<code>s.findDownstream(fromNodes)</code>	Collection	Searches from the supplied list of nodes and returns the set of nodes downstream of the supplied nodes. The returned list includes the originally supplied nodes.
<code>s.findUpstream(fromNodes)</code>	Collection	Searches from the supplied list of nodes and returns the set of nodes upstream of the supplied nodes. The returned list includes the originally supplied nodes.
<code>s.findProcessorForID(String id, boolean recursive)</code>	Node	Returns the node with the supplied ID or <code>None</code> if no such node exists. If the recursive flag is <code>true</code> , then any composite nodes within this diagram are also searched.

As an example, if a stream contained a single Filter node that the script needed to access, the Filter node can be found by using the following script:

```
stream = modeler.script.stream()
node = stream.findByType("filter", None)
...
```

Alternatively, if the ID of the node (as shown on the Annotations tab of the node dialog box) is known, the ID can be used to find the node, for example:

```
stream = modeler.script.stream()
node = stream.findById("id32FJT71G2") # the filter node ID
...
```

Setting properties

Nodes, streams, models, and outputs all have properties that can be accessed and, in most cases, set. Properties are typically used to modify the behavior or appearance of the object. The methods that are available for accessing and setting object properties are summarized in the following table.

Table 1. Methods for accessing and setting object properties

Method	Return type	Description
<code>p.getPropertyValue(propertyName)</code>	Object	Returns the value of the named property or <code>None</code> if no such property exists.
<code>p.setPropertyValue(propertyName, value)</code>	Not applicable	Sets the value of the named property.
<code>p.setPropertyValues(properties)</code>	Not applicable	Sets the values of the named properties. Each entry in the properties map consists of a key that represents the property name and the value that should be assigned to that property.
<code>p.getKeyedPropertyValue(propertyName, keyName)</code>	Object	Returns the value of the named property and associated key or <code>None</code> if no such property or key exists.
<code>p.setKeyedPropertyValue(propertyName, keyName, value)</code>	Not applicable	Sets the value of the named property and key.

For example, if you wanted to set the value of a Variable File node at the start of a stream, you can use the following script:

```
stream = modeler.script.stream()
node = stream.findByType("variablefile", None)
node.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")
...
```

Alternatively, you might want to filter a field from a Filter node. In this case, the value is also keyed on the field name, for example:

```
stream = modeler.script.stream()
# Locate the filter node ...
node = stream.findByType("filter", None)
# ... and filter out the "Na" field
node.setKeyedPropertyValue("include", "Na", False)
```

Related information

- [Referencing existing nodes](#)
- [Finding nodes](#)

Creating nodes and modifying streams

In some situations, you might want to add new nodes to existing streams. Adding nodes to existing streams typically involves the following tasks:

1. Creating the nodes.
 2. Linking the nodes into the existing stream flow.
- [Creating nodes](#)
 - [Linking and unlinking nodes](#)
 - [Importing, replacing, and deleting nodes](#)
 - [Traversing through nodes in a stream](#)

Related information

- [Creating nodes](#)
- [Linking and unlinking nodes](#)
- [Importing, replacing, and deleting nodes](#)
- [Traversing through nodes in a stream](#)
- [Types of scripts](#)
- [Executing a stream](#)
- [The scripting context](#)
- [Referencing existing nodes](#)
- [Getting information about nodes](#)

Creating nodes

Streams provide a number of ways of creating nodes. These methods are summarized in the following table.

Table 1. Methods for creating nodes

Method	Return type	Description
<code>s.create(nodeType, name)</code>	Node	Creates a node of the specified type and adds it to the specified stream.
<code>s.createAt(nodeType, name, x, y)</code>	Node	Creates a node of the specified type and adds it to the specified stream at the specified location. If either x < 0 or y < 0, the location is not set.
<code>s.createModelApplier(modelOutput, name)</code>	Node	Creates a model applier node that is derived from the supplied model output object.

For example, to create a new Type node in a stream you can use the following script:

```
stream = modeler.script.stream()
# Create a new type node
node = stream.create("type", "My Type")
```

Related information

- [Creating nodes and modifying streams](#)
- [Linking and unlinking nodes](#)
- [Importing, replacing, and deleting nodes](#)
- [Traversing through nodes in a stream](#)

Linking and unlinking nodes

When a new node is created within a stream, it must be connected into a sequence of nodes before it can be used. Streams provide a number of methods for linking and unlinking nodes. These methods are summarized in the following table.

Table 1. Methods for linking and unlinking nodes

Method	Return type	Description
<code>s.link(source, target)</code>	Not applicable	Creates a new link between the source and the target nodes.
<code>s.link(source, targets)</code>	Not applicable	Creates new links between the source node and each target node in the supplied list.
<code>s.linkBetween(inserted, source, target)</code>	Not applicable	Connects a node between two other node instances (the source and target nodes) and sets the position of the inserted node to be between them. Any direct link between the source and target nodes is removed first.
<code>s.linkPath(path)</code>	Not applicable	Creates a new path between node instances. The first node is linked to the second, the second is linked to the third, and so on.

Method	Return type	Description
<code>s.unlink(source, target)</code>	Not applicable	Removes any direct link between the source and the target nodes.
<code>s.unlink(source, targets)</code>	Not applicable	Removes any direct links between the source node and each object in the targets list.
<code>s.unlinkPath(path)</code>	Not applicable	Removes any path that exists between node instances.
<code>s.disconnect(node)</code>	Not applicable	Removes any links between the supplied node and any other nodes in the specified stream.
<code>s.isValidLink(source, target)</code>	<code>boolean</code>	Returns <code>True</code> if it would be valid to create a link between the specified source and target nodes. This method checks that both objects belong to the specified stream, that the source node can supply a link and the target node can receive a link, and that creating such a link will not cause a circularity in the stream.

The example script that follows performs these five tasks:

1. Creates a Variable File input node, a Filter node, and a Table output node.
2. Connects the nodes together.
3. Sets the file name on the Variable File input node.
4. Filters the field "Drug" from the resulting output.
5. Executes the Table node.

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", "My File Input ", 96, 64)
filternode = stream.createAt("filter", "Filter", 192, 64)
tablenode = stream.createAt("table", "Table", 288, 64)
stream.link(filenode, filternode)
stream.link(filternode, tablenode)
filenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
filternode.setKeyedPropertyValue("include", "Drug", False)
results = []
tablenode.run(results)
```

Importing, replacing, and deleting nodes

As well as creating and connecting nodes, it is often necessary to replace and delete nodes from the stream. The methods that are available for importing, replacing and deleting nodes are summarized in the following table.

Table 1. Methods for importing, replacing, and deleting nodes

Method	Return type	Description
<code>s.replace(originalNode, replacementNode, discardOriginal)</code>	Not applicable	Replaces the specified node from the specified stream. Both the original node and replacement node must be owned by the specified stream.
<code>s.insert(source, nodes, newIDs)</code>	List	Inserts copies of the nodes in the supplied list. It is assumed that all nodes in the supplied list are contained within the specified stream. The <code>newIDs</code> flag indicates whether new IDs should be generated for each node, or whether the existing ID should be copied and used. It is assumed that all nodes in a stream have a unique ID, so this flag must be set to <code>True</code> if the source stream is the same as the specified stream. The method returns the list of newly inserted nodes, where the order of the nodes is undefined (that is, the ordering is not necessarily the same as the order of the nodes in the input list).
<code>s.delete(node)</code>	Not applicable	Deletes the specified node from the specified stream. The node must be owned by the specified stream.
<code>s.deleteAll(nodes)</code>	Not applicable	Deletes all the specified nodes from the specified stream. All nodes in the collection must belong to the specified stream.
<code>s.clear()</code>	Not applicable	Deletes all nodes from the specified stream.

Related information

- [Creating nodes and modifying streams](#)
- [Creating nodes](#)
- [Linking and unlinking nodes](#)
- [Traversing through nodes in a stream](#)

Traversing through nodes in a stream

A common requirement is to identify nodes that are either upstream or downstream of a particular node. The stream provides a number of methods that can be used to identify these nodes. These methods are summarized in the following table.

Table 1. Methods to identify upstream and downstream nodes

Method	Return type	Description
<code>s.iterator()</code>	Iterator	Returns an iterator over the node objects that are contained in the specified stream. If the stream is modified between calls of the <code>next()</code> function, the behavior of the iterator is undefined.
<code>s.predecessorAt(node, index)</code>	Node	Returns the specified immediate predecessor of the supplied node or <code>None</code> if the index is out of bounds.
<code>s.predecessorCount(node)</code>	<code>int</code>	Returns the number of immediate predecessors of the supplied node.
<code>s.predecessors(node)</code>	List	Returns the immediate predecessors of the supplied node.
<code>s.successorAt(node, index)</code>	Node	Returns the specified immediate successor of the supplied node or <code>None</code> if the index is out of bounds.
<code>s.successorCount(node)</code>	<code>int</code>	Returns the number of immediate successors of the supplied node.
<code>s.successors(node)</code>	List	Returns the immediate successors of the supplied node.

Related information

- [Creating nodes and modifying streams](#)
- [Creating nodes](#)
- [Linking and unlinking nodes](#)
- [Importing, replacing, and deleting nodes](#)

Clearing, or removing, items

Legacy scripting supports various uses of the `clear` command, for example:

- `clear outputs` To delete all output items from the manager palette.
- `clear generated palette` To clear all model nuggets from the Models palette.
- `clear stream` To remove the contents of a stream.

Python scripting supports a similar set of functions; the `removeAll()` command is used to clear the Streams, Outputs, and Models managers. For example:

- To clear the Streams manager:

```
session = modeler.script.session()
session.getStreamManager().removeAll()
```

- To clear the Outputs manager:

```
session = modeler.script.session()
session.getDocumentOutputManager().removeAll()
```

- To clear the Models manager:

```
session = modeler.script.session()
session.getModelOutputManager().removeAll()
```

Getting information about nodes

Nodes fall into a number of different categories such as data import and export nodes, model building nodes, and other types of nodes. Every node provides a number of methods that can be used to find out information about the node.

The methods that can be used to obtain the ID, name, and label of a node are summarized in the following table.

Table 1. Methods to obtain the ID, name, and label of a node

Method	Return type	Description
<code>n.getLabel()</code>	<code>string</code>	Returns the display label of the specified node. The label is the value of the property <code>custom_name</code> only if that property is a non-empty string and the <code>use_custom_name</code> property is not set; otherwise, the label is the value of <code>getName()</code> .

Method	Return type	Description
<code>n.setLabel(label)</code>	Not applicable	Sets the display label of the specified node. If the new label is a non-empty string it is assigned to the property <code>custom_name</code> , and <code>False</code> is assigned to the property <code>use_custom_name</code> so that the specified label takes precedence; otherwise, an empty string is assigned to the property <code>custom_name</code> and <code>True</code> is assigned to the property <code>use_custom_name</code> .
<code>n.getName()</code>	<code>string</code>	Returns the name of the specified node.
<code>n.getID()</code>	<code>string</code>	Returns the ID of the specified node. A new ID is created each time a new node is created. The ID is persisted with the node when it is saved as part of a stream so that when the stream is opened, the node IDs are preserved. However, if a saved node is inserted into a stream, the inserted node is considered to be a new object and will be allocated a new ID.

Methods that can be used to obtain other information about a node are summarized in the following table.

Table 2. Methods for obtaining information about a node

Method	Return type	Description
<code>n.getTypeName()</code>	<code>string</code>	Returns the scripting name of this node. This is the same name that could be used to create a new instance of this node.
<code>n.isInitial()</code>	<code>Boolean</code>	Returns <code>True</code> if this is an <i>initial</i> node, that is one that occurs at the start of a stream.
<code>n.isInline()</code>	<code>Boolean</code>	Returns <code>True</code> if this is an <i>in-line</i> node, that is one that occurs mid-stream.
<code>n.isTerminal()</code>	<code>Boolean</code>	Returns <code>True</code> if this is a <i>terminal</i> node, that is one that occurs at the end of a stream.
<code>n.getXPosition()</code>	<code>int</code>	Returns the x position offset of the node in the stream.
<code>n.getYPosition()</code>	<code>int</code>	Returns the y position offset of the node in the stream.
<code>n.setXYPosition(x, y)</code>	Not applicable	Sets the position of the node in the stream.
<code>n.setPositionBetween(source, target)</code>	Not applicable	Sets the position of the node in the stream so that it is positioned between the supplied nodes.
<code>n.isCacheEnabled()</code>	<code>Boolean</code>	Returns <code>True</code> if the cache is enabled; returns <code>False</code> otherwise.
<code>n.setCacheEnabled(val)</code>	Not applicable	Enables or disables the cache for this object. If the cache is full and the caching becomes disabled, the cache is flushed.
<code>n.isCacheFull()</code>	<code>Boolean</code>	Returns <code>True</code> if the cache is full; returns <code>False</code> otherwise.
<code>n.flushCache()</code>	Not applicable	Flushes the cache of this node. Has no affect if the cache is not enabled or is not full.

Related information

- [Types of scripts](#)
- [Executing a stream](#)
- [The scripting context](#)
- [Referencing existing nodes](#)
- [Creating nodes and modifying streams](#)

The Scripting API

- [Introduction to the Scripting API](#)
- [Example 1: searching for nodes using a custom filter](#)
- [Example 2: allowing users to obtain directory or file information based on their privileges](#)
- [Metadata: Information about data](#)
- [Accessing Generated Objects](#)
- [Handling errors](#)
- [Stream, Session, and SuperNode Parameters](#)
- [Global Values](#)
- [Working with Multiple Streams: Standalone Scripts](#)

Introduction to the Scripting API

The Scripting API provides access to a wide range of SPSS® Modeler functionality. All the methods described so far are part of the API and can be accessed implicitly within the script without further imports. However, if you want to reference the API classes, you must import the API explicitly with the following statement:

```
import modeler.api
```

This import statement is required by many of the Scripting API examples.

Related information

- [Example 1: searching for nodes using a custom filter](#)
 - [Metadata: Information about data](#)
 - [Accessing Generated Objects](#)
 - [Handling errors](#)
 - [Stream, Session, and SuperNode Parameters](#)
 - [Global Values](#)
 - [Working with Multiple Streams: Standalone Scripts](#)
-

Example 1: searching for nodes using a custom filter

The section [Finding nodes](#) included an example of searching for a node in a stream using the type name of the node as the search criterion. In some situations, a more generic search is required and this can be implemented using the `NodeFilter` class and the stream `findAll()` method. This kind of search involves the following two steps:

1. Creating a new class that extends `NodeFilter` and that implements a custom version of the `accept()` method.
2. Calling the stream `findAll()` method with an instance of this new class. This returns all nodes that meet the criteria defined in the `accept()` method.

The following example shows how to search for nodes in a stream that have the node cache enabled. The returned list of nodes could be used to either flush or disable the caches of these nodes.

```
import modeler.api

class CacheFilter(modeler.api.NodeFilter):
    """A node filter for nodes with caching enabled"""
    def accept(this, node):
        return node.isCacheEnabled()

cachingnodes = modeler.script.stream().findAll(CacheFilter(), False)
```

Example 2: allowing users to obtain directory or file information based on their privileges

To avoid the PSAPI being opened to users, a method called `session.getServerFileSystem()` can be used via calling the PSAPI function to create a file system object.

The following example shows how to allow a user to get directory or file information based on the privileges of the user that connects to the IBM® SPSS® Modeler Server.

```
import modeler.api
stream = modeler.script.stream()
sourceNode = stream.findByID('')
session = modeler.script.session()
fileSystem = session.getServerFileSystem()
parameter = stream.getParameterValue('VPATH')
serverDirectory = fileSystem.getServerFile(parameter)
files = fileSystem.GetFiles(serverDirectory)
for f in files:
    if f.isDirectory():
        print 'Directory:'
    else:
        print 'File:'
        sourceNode.setPropertyValue('full_filename',f.getPath())
        break
    print f.getName(),f.getPath()
stream.execute()
```

Metadata: Information about data

Because nodes are connected together in a stream, information about the columns or fields that are available at each node is available. For example, in the Modeler UI, this allows you to select which fields to sort or aggregate by. This information is called the data model.

Scripts can also access the data model by looking at the fields coming into or out of a node. For some nodes, the input and output data models are the same, for example a Sort node simply reorders the records but doesn't change the data model. Some, such as the Derive node, can add new fields. Others, such as the Filter node can rename or remove fields.

In the following example, the script takes the standard IBM® SPSS® Modeler druglearn.str stream, and for each field, builds a model with one of the input fields dropped. It does this by:

1. Accessing the output data model from the Type node.
2. Looping through each field in the output data model.
3. Modifying the Filter node for each input field.
4. Changing the name of the model being built.
5. Running the model build node.

Note: Before running the script in the druglean.str stream, remember to set the scripting language to Python (the stream was created in a previous version of IBM SPSS Modeler so the stream scripting language is set to Legacy).

```
import modeler.api

stream = modeler.script.stream()
filternode = stream.findByType("filter", None)
typenode = stream.findByType("type", None)
c50node = stream.findByType("c50", None)
# Always use a custom model name
c50node.setPropertyValue("use_model_name", True)

lastRemoved = None
fields = typenode.getOutputDataModel()
for field in fields:
    # If this is the target field then ignore it
    if field.getModelingRole() == modeler.api.ModelingRole.OUT:
        continue

    # Re-enable the field that was most recently removed
    if lastRemoved != None:
        filternode.setKeyedPropertyValue("include", lastRemoved, True)

    # Remove the field
    lastRemoved = field.getColumnName()
    filternode.setKeyedPropertyValue("include", lastRemoved, False)

    # Set the name of the new model then run the build
    c50node.setPropertyValue("model_name", "Exclude " + lastRemoved)
    c50node.run([])
```

The DataModel object provides a number of methods for accessing information about the fields or columns within the data model. These methods are summarized in the following table.

Table 1. DataModel object methods for accessing information about fields or columns

Method	Return type	Description
d.getColumnCount()	int	Returns the number of columns in the data model.
d.columnIterator()	Iterator	Returns an iterator that returns each column in the "natural" insert order. The iterator returns instances of Column.
d.nameIterator()	Iterator	Returns an iterator that returns the name of each column in the "natural" insert order.
d.contains(name)	Boolean	Returns True if a column with the supplied name exists in this DataModel, False otherwise.
d.getColumn(name)	Column	Returns the column with the specified name.
d.getColumnGroup(name)	ColumnGroup	Returns the named column group or None if no such column group exists.
d.getColumnGroupCount()	int	Returns the number of column groups in this data model.
d.columnGroupIterator()	Iterator	Returns an iterator that returns each column group in turn.
d.toArray()	Column[]	Returns the data model as an array of columns. The columns are ordered in their "natural" insert order.

Each field (Column object) includes a number of methods for accessing information about the column. The table below shows a selection of these.

Table 2. Column object methods for accessing information about the column

Method	Return type	Description
c.getColumnName()	string	Returns the name of the column.

Method	Return type	Description
<code>c.getColumnLabel()</code>	<code>string</code>	Returns the label of the column or an empty string if there is no label associated with the column.
<code>c.getMeasureType()</code>	<code>MeasureType</code>	Returns the measure type for the column.
<code>c.getStorageType()</code>	<code>StorageType</code>	Returns the storage type for the column.
<code>c.isMeasureDiscrete()</code>	<code>Boolean</code>	Returns <code>True</code> if the column is discrete. Columns that are either a set or a flag are considered discrete.
<code>c.isModelOutputColumn()</code>	<code>Boolean</code>	Returns <code>True</code> if the column is a model output column.
<code>c.isStorageDatetime()</code>	<code>Boolean</code>	Returns <code>True</code> if the column's storage is a time, date or timestamp value.
<code>c.isStorageNumeric()</code>	<code>Boolean</code>	Returns <code>True</code> if the column's storage is an integer or a real number.
<code>c.isValidValue(value)</code>	<code>Boolean</code>	Returns <code>True</code> if the specified value is valid for this storage, and <code>valid</code> when the valid column values are known.
<code>c.getModelingRole()</code>	<code>ModelingRole</code>	Returns the modeling role for the column.
<code>c.getSetValues()</code>	<code>Object[]</code>	Returns an array of valid values for the column, or <code>None</code> if either the values are not known or the column is not a set.
<code>c.getValueLabel(value)</code>	<code>string</code>	Returns the label for the value in the column, or an empty string if there is no label associated with the value.
<code>c.getFalseFlag()</code>	<code>Object</code>	Returns the "false" indicator value for the column, or <code>None</code> if either the value is not known or the column is not a flag.
<code>c.getTrueFlag()</code>	<code>Object</code>	Returns the "true" indicator value for the column, or <code>None</code> if either the value is not known or the column is not a flag.
<code>c.getLowerBound()</code>	<code>Object</code>	Returns the lower bound value for the values in the column, or <code>None</code> if either the value is not known or the column is not continuous.
<code>c.getUpperBound()</code>	<code>Object</code>	Returns the upper bound value for the values in the column, or <code>None</code> if either the value is not known or the column is not continuous.

Note that most of the methods that access information about a column have equivalent methods defined on the DataModel object itself. For example the two following statements are equivalent:

```
dataModel.getColumn("someName").getModelingRole()
dataModel.getModelingRole("someName")
```

Related information

- [Introduction to the Scripting API](#)
- [Example 1: searching for nodes using a custom filter](#)
- [Accessing Generated Objects](#)
- [Handling errors](#)
- [Stream, Session, and SuperNode Parameters](#)
- [Global Values](#)
- [Working with Multiple Streams: Standalone Scripts](#)

Accessing Generated Objects

Executing a stream typically involves producing additional output objects. These additional objects might be a new model, or a piece of output that provides information to be used in subsequent executions.

In the example below, the druglearn.str stream is used again as the starting point for the stream. In this example, all nodes in the stream are executed and the results are stored in a list. The script then loops through the results, and any model outputs that result from the execution are saved as an IBM® SPSS® Modeler model (.gm) file, and the model is PMML exported.

```
import modeler.api

stream = modeler.script.stream()

# Set this to an existing folder on your system.
# Include a trailing directory separator
modelFolder = "C:/temp/models/"

# Execute the stream
models = []
stream.runAll(models)
```

```

# Save any models that were created
taskrunner = modeler.script.session().getTaskRunner()
for model in models:
    # If the stream execution built other outputs then ignore them
    if not(isinstance(model, modeler.api.ModelOutput)):
        continue

    label = model.getLabel()
    algorithm = model.getModelDetail().getAlgorithmName()

    # save each model...
    modelFile = modelFolder + label + algorithm + ".gm"
    taskrunner.saveModelToFile(model, modelFile)

    # ...and export each model PMML...
    modelFile = modelFolder + label + algorithm + ".xml"
    taskrunner.exportModelToFile(model, modelFile, modeler.api.FileFormat.XML)

```

The task runner class provides a convenient way running various common tasks. The methods that are available in this class are summarized in the following table.

Table 1. Methods of the task runner class for performing common tasks

Method	Return type	Description
t.createStream(name, autoConnect, autoManage)	Stream	Creates and returns a new stream. Note that code that must create streams privately without making them visible to the user should set the autoManage flag to False .
t.exportDocumentToFile(documentOutput, filename, fileFormat)	Not applicable	Exports the stream description to a file using the specified file format.
t.exportModelToFile(modelOutput, filename, fileFormat)	Not applicable	Exports the model to a file using the specified file format.
t.exportStreamToFile(stream, filename, fileFormat)	Not applicable	Exports the stream to a file using the specified file format.
t.insertNodeFromFile(fileName, diagram)	Node	Reads and returns a node from the specified file, inserting it into the supplied diagram. Note that this can be used to read both Node and SuperNode objects.
t.openDocumentFromFile(fileName, autoManage)	DocumentOutput	Reads and returns a document from the specified file.
t.openModelFromFile(fileName, autoManage)	ModelOutput	Reads and returns a model from the specified file.
t.openStreamFromFile(fileName, autoManage)	Stream	Reads and returns a stream from the specified file.
t.saveDocumentToFile(documentOutput, filename)	Not applicable	Saves the document to the specified file location.
t.saveModelToFile(modelOutput, filename)	Not applicable	Saves the model to the specified file location.
t.saveStreamToFile(stream, filename)	Not applicable	Saves the stream to the specified file location.

Handling errors

The Python language provides error handling via the `try...except` code block. This can be used within scripts to trap exceptions and handle problems that would otherwise cause the script to terminate.

In the example script below, an attempt is made to retrieve a model from a IBM® SPSS® Collaboration and Deployment Services Repository. This operation can cause an exception to be thrown, for example, the repository login credentials might not have been set up correctly, or the repository path is wrong. In the script, this may cause a `ModelerException` to be thrown (all exceptions that are generated by IBM SPSS Modeler are derived from `modeler.api.ModelerException`).

```

import modeler.api

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # print goes to the Modeler UI script panel Debug tab
    print "Everything OK"
except modeler.api.ModelerException, e:
    print "An error occurred:", e.getMessage()

```

Note: Some scripting operations may cause standard Java exceptions to be thrown; these are not derived from `ModelerException`. In order to catch these exceptions, an additional except block can be used to catch all Java exceptions, for example:

```

import modeler.api
import java.lang.Exception

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # print goes to the Modeler UI script panel Debug tab
    print "Everything OK"
except modeler.api.ModelerException, e:
    print "An error occurred:", e.getMessage()
except java.lang.Exception, e:
    print "A Java exception occurred:", e.getMessage()

```

Related information

- [Introduction to the Scripting API](#)
 - [Example 1: searching for nodes using a custom filter](#)
 - [Metadata: Information about data](#)
 - [Accessing Generated Objects](#)
 - [Stream, Session, and SuperNode Parameters](#)
 - [Global Values](#)
 - [Working with Multiple Streams: Standalone Scripts](#)
-

Stream, Session, and SuperNode Parameters

Parameters provide a useful way of passing values at runtime, rather than hard coding them directly in a script. Parameters and their values are defined in the same as way for streams, that is, as entries in the parameters table of a stream or SuperNode, or as parameters on the command line. The Stream and SuperNode classes implement a set of functions defined by the ParameterProvider object as shown in the following table. Session provides a `getParameters()` call which returns an object that defines those functions.

Table 1. Functions defined by the ParameterProvider object

Method	Return type	Description
<code>p.getParameterIterator()</code>	Iterator	Returns an iterator of parameter names for this object.
<code>p.getParameterDefinition(parameterName)</code>	ParameterDefinition	Returns the parameter definition for the parameter with the specified name, or <code>None</code> if no such parameter exists in this provider. The result may be a snapshot of the definition at the time the method was called and need not reflect any subsequent modifications made to the parameter through this provider.
<code>p.getParameterLabel(parameterName)</code>	string	Returns the label of the named parameter, or <code>None</code> if no such parameter exists.
<code>p.setParameterLabel(parameterName, label)</code>	Not applicable	Sets the label of the named parameter.
<code>p.getParameterStorage(parameterName)</code>	ParameterStorage	Returns the storage of the named parameter, or <code>None</code> if no such parameter exists.
<code>p.setParameterStorage(parameterName, storage)</code>	Not applicable	Sets the storage of the named parameter.
<code>p.getParameterType(parameterName)</code>	ParameterType	Returns the type of the named parameter, or <code>None</code> if no such parameter exists.
<code>p.setParameterType(parameterName, type)</code>	Not applicable	Sets the type of the named parameter.
<code>p.getParameterValue(parameterName)</code>	Object	Returns the value of the named parameter, or <code>None</code> if no such parameter exists.
<code>p.setParameterValue(parameterName, value)</code>	Not applicable	Sets the value of the named parameter.

In the following example, the script aggregates some Telco data to find which region has the lowest average income data. A stream parameter is then set with this region. That stream parameter is then used in a Select node to exclude that region from the data, before a churn model is built on the remainder.

The example is artificial because the script generates the Select node itself and could therefore have generated the correct value directly into the Select node expression. However, streams are typically pre-built, so setting parameters in this way provides a useful example.

The first part of the example script creates the stream parameter that will contain the region with the lowest average income. The script also creates the nodes in the aggregation branch and the model building branch, and connects them together.

```

import modeler.api

stream = modeler.script.stream()

# Initialize a stream parameter
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# First create the aggregation branch to compute the average income per region
statisticsimportnode = stream.createAt("statisticsimport", "SPSS File", 114, 142)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/telco.sav")
statisticsimportnode.setPropertyValue("use_field_format_for_storage", True)

aggregatenode = modeler.script.stream().createAt("aggregate", "Aggregate", 294, 142)
aggregatenode.setPropertyValue("keys", ["region"])
aggregatenode.setKeyedPropertyValue("aggregates", "income", ["Mean"])

tablenode = modeler.script.stream().createAt("table", "Table", 462, 142)

stream.link(statisticsimportnode, aggregatenode)
stream.link(aggregatenode, tablenode)

selectnode = stream.createAt("select", "Select", 210, 232)
selectnode.setPropertyValue("mode", "Discard")
# Reference the stream parameter in the selection
selectnode.setPropertyValue("condition", "'region' = '$P-LowestRegion'")

typenode = stream.createAt("type", "Type", 366, 232)
typenode.setKeyedPropertyValue("direction", "churn", "Target")

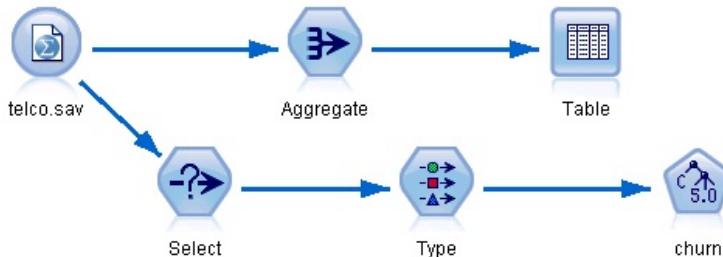
c50node = stream.createAt("c50", "C5.0", 534, 232)

stream.link(statisticsimportnode, selectnode)
stream.link(selectnode, typenode)
stream.link(typenode, c50node)

```

The example script creates the following stream.

Figure 1. Stream that results from the example script



The following part of the example script executes the Table node at the end of the aggregation branch.

```

# First execute the table node
results = []
tablenode.run(results)

```

The following part of the example script accesses the table output that was generated by the execution of the Table node. The script then iterates through rows in the table, looking for the region with the lowest average income.

```

# Running the table node should produce a single table as output
table = results[0]

# table output contains a RowSet so we can access values as rows and columns
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

# From the way the aggregate node is defined, the first column
# contains the region and the second contains the average income
row = 0
rowcount = rowset.getRowCount()
while row < rowcount:
    if rowset.getValueAt(row, 1) < min_income:
        min_income = rowset.getValueAt(row, 1)

```

```

        min_region = rowset.getValueAt(row, 0)
        row += 1

```

The following part of the script uses the region with the lowest average income to set the "LowestRegion" stream parameter that was created earlier. The script then runs the model builder with the specified region excluded from the training data.

```

# Check that a value was assigned
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)

# Finally run the model builder with the selection criteria
c50node.run([])

```

The complete example script is shown below.

```

import modeler.api

stream = modeler.script.stream()

# Create a stream parameter
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# First create the aggregation branch to compute the average income per region
statisticsimportnode = stream.createAt("statisticsimport", "SPSS File", 114, 142)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/telco.sav")
statisticsimportnode.setPropertyValue("use_field_format_for_storage", True)

aggregatenode = modeler.script.stream().createAt("aggregate", "Aggregate", 294, 142)
aggregatenode.setPropertyValue("keys", ["region"])
aggregatenode.setKeyedPropertyValue("aggregates", "income", ["Mean"])

tablenode = modeler.script.stream().createAt("table", "Table", 462, 142)

stream.link(statisticsimportnode, aggregatenode)
stream.link(aggregatenode, tablenode)

selectnode = stream.createAt("select", "Select", 210, 232)
selectnode.setPropertyValue("mode", "Discard")
# Reference the stream parameter in the selection
selectnode.setPropertyValue("condition", "'region' = '$P-LowestRegion'")

typenode = stream.createAt("type", "Type", 366, 232)
typenode.setKeyedPropertyValue("direction", "churn", "Target")

c50node = stream.createAt("c50", "C5.0", 534, 232)

stream.link(statisticsimportnode, selectnode)
stream.link(selectnode, typenode)
stream.link(typenode, c50node)

# First execute the table node
results = []
tablenode.run(results)

# Running the table node should produce a single table as output
table = results[0]

# table output contains a RowSet so we can access values as rows and columns
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

# From the way the aggregate node is defined, the first column
# contains the region and the second contains the average income
row = 0
rowcount = rowset.getRowCount()
while row < rowcount:
    if rowset.getValueAt(row, 1) < min_income:
        min_income = rowset.getValueAt(row, 1)
        min_region = rowset.getValueAt(row, 0)
    row += 1

# Check that a value was assigned
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)

```

```
# Finally run the model builder with the selection criteria
c50node.run([])
```

Global Values

Global values are used to compute various summary statistics for specified fields. These summary values can be accessed anywhere within the stream. Global values are similar to stream parameters in that they are accessed by name through the stream. They are different from stream parameters in that the associated values are updated automatically when a Set Globals node is run, rather than being assigned by scripting or from the command line. The global values for a stream are accessed by calling the stream's `getGlobalValues()` method.

The `GlobalValues` object defines the functions that are shown in the following table.

Table 1. Functions that are defined by the `GlobalValues` object

Method	Return type	Description
<code>g.fieldNameIterator()</code>	Iterator	Returns an iterator for each field name with at least one global value.
<code>g.getValue(type, fieldName)</code>	Object	Returns the global value for the specified type and field name, or <code>None</code> if no value can be located. The returned value is generally expected to be a number, although future functionality may return different value types.
<code>g.getValues(fieldName)</code>	Map	Returns a map containing the known entries for the specified field name, or <code>None</code> if there are no existing entries for the field.

`GlobalValues.Type` defines the type of summary statistics that are available. The following summary statistics are available:

- **MAX**: the maximum value of the field.
- **MEAN**: the mean value of the field.
- **MIN**: the minimum value of the field.
- **STDDEV**: the standard deviation of the field.
- **SUM**: the sum of the values in the field.

For example, the following script accesses the mean value of the "income" field, which is computed by a Set Globals node:

```
import modeler.api

globals = modeler.script.stream().getGlobalValues()
mean_income = globals.getValue(modeler.api.GlobalValues.Type.MEAN, "income")
```

Related information

- [Introduction to the Scripting API](#)
- [Example 1: searching for nodes using a custom filter](#)
- [Metadata: Information about data](#)
- [Accessing Generated Objects](#)
- [Handling errors](#)
- [Stream, Session, and SuperNode Parameters](#)
- [Working with Multiple Streams: Standalone Scripts](#)

Working with Multiple Streams: Standalone Scripts

To work with multiple streams, a standalone script must be used. The standalone script can be edited and run within the IBM® SPSS® Modeler UI or passed as a command line parameter in batch mode.

The following standalone script opens two streams. One of these streams builds a model, while the second stream plots the distribution of the predicted values.

```
# Change to the appropriate location for your system
demosDir = "C:/Program Files/IBM/SPSS/Modeler/18.4.0/DEMONS/streams/"

session = modeler.script.session()
tasks = session.getTaskRunner()

# Open the model build stream, locate the C5.0 node and run it
buildstream = tasks.openStreamFromFile(demosDir + "druglearn.str", True)
c50node = buildstream.findByName("c50", None)
results = []
c50node.run(results)
```

```

# Now open the plot stream, find the Na_to_K derive and the histogram
plotstream = tasks.openStreamFromFile(demosDir + "drugplot.str", True)
derivenode = plotstream.findByType("derive", None)
histogramnode = plotstream.findByType("histogram", None)

# Create a model applier node, insert it between the derive and histogram nodes
# then run the histogram
applyc50 = plotstream.createModelApplier(results[0], results[0].getName())
applyc50.setPositionBetween(derivenode, histogramnode)
plotstream.linkBetween(applyc50, derivenode, histogramnode)
histogramnode.setPropertyValue("color_field", "$C-Drug")
histogramnode.run([])

# Finally, tidy up the streams
buildstream.close()
plotstream.close()

```

The following example shows how you can also iterate over the open streams (all the streams open in the Streams tab). Note that this is only supported in standalone scripts.

```

for stream in modeler.script.streams():
    print stream.getName()

```

Related information

- [Introduction to the Scripting API](#)
- [Example 1: searching for nodes using a custom filter](#)
- [Metadata: Information about data](#)
- [Accessing Generated Objects](#)
- [Handling errors](#)
- [Stream, Session, and SuperNode Parameters](#)
- [Global Values](#)

Scripting tips

This section provides an overview of tips and techniques for using scripts, including modifying stream execution, using an encoded password in a script, and accessing objects in the IBM® SPSS® Collaboration and Deployment Services Repository.

- [Modifying stream execution](#)
- [Looping through nodes](#)
- [Accessing Objects in the IBM SPSS Collaboration and Deployment Services Repository](#)
- [Generating an encoded password](#)
- [Script checking](#)
- [Scripting from the command line](#)
- [Compatibility with previous releases](#)
- [Accessing stream execution results](#)

Modifying stream execution

When a stream is run, its terminal nodes are executed in an order optimized for the default situation. In some cases, you may prefer a different execution order. To modify the execution order of a stream, complete the following steps from the Execution tab of the stream properties dialog box:

1. Begin with an empty script.
2. Click the Append default script button on the toolbar to add the default stream script.
3. Change the order of statements in the default stream script to the order in which you want statements to be executed.

Looping through nodes

You can use a `for` loop to loop through all of the nodes in a stream. For example, the following two script examples loop through all nodes and changes field names in any Filter nodes to upper case.

This scripts can be used in any stream that has a Filter node, even if no fields are actually filtered. Simply add a Filter node that passes all fields in order to change field names to upper case across the board.

```

# Alternative 1: using the data model nameIterator() function
stream = modeler.script.stream()
for node in stream.iterator():
    if (node.getTypeName() == "filter"):
        # nameIterator() returns the field names
        for field in node.getInputDataModel().nameIterator():
            newname = field.upper()
            node.setKeyedPropertyValue("new_name", field, newname)

# Alternative 2: using the data model iterator() function
stream = modeler.script.stream()
for node in stream.iterator():
    if (node.getTypeName() == "filter"):
        # iterator() returns the field objects so we need
        # to call getColumnNames() to get the name
        for field in node.getInputDataModel().iterator():
            newname = field.getColumnName().upper()
            node.setKeyedPropertyValue("new_name", field.getColumnName(), newname)

```

The script loops through all nodes in the current stream, and checks whether each node is a Filter. If so, the script loops through each field in the node and uses either the `field.upper()` or `field.getColumnName().upper()` function to change the name to upper case.

Accessing Objects in the IBM SPSS Collaboration and Deployment Services Repository

If you have a license for the IBM® SPSS® Collaboration and Deployment Services Repository, you can store and retrieve objects from the repository by using script commands. Use the repository to manage the lifecycle of data mining models and related predictive objects in the context of enterprise applications, tools, and solutions.

Connecting to the IBM SPSS Collaboration and Deployment Services Repository

To access the repository, you must first set up a valid connection to it, either through the Tools menu of the SPSS Modeler user interface or through the command line. For more information, see [IBM SPSS Collaboration and Deployment Services Repository Connection Arguments](#).

Getting access to the repository

The repository can be accessed from the session, for example:

```
repo = modeler.script.session().getRepository()
```

Retrieving objects from the repository

Within a script, use the `retrieve*` functions to access various objects, including streams, models, output, and nodes. A summary of the retrieval functions is shown in the following table.

Table 1. Retrieve scripting functions

Object Type	Repository Function
Stream	<code>repo.retrieveStream(String path, String version, String label, Boolean autoManage)</code>
Model	<code>repo.retrieveModel(String path, String version, String label, Boolean autoManage)</code>
Output	<code>repo.retrieveDocument(String path, String version, String label, Boolean autoManage)</code>
Node	<code>repo.retrieveProcessor(String path, String version, String label, ProcessorDiagram diagram)</code>

For example, you can retrieve a stream from the repository with the following function:

```
stream = repo.retrieveStream("/projects/retention/risk_score.str", None, "production", True)
```

This example retrieves the `risk_score.str` stream from the specified folder. The label `production` identifies which version of the stream to retrieve, and the last parameter specifies that SPSS Modeler is to manage the stream (for example, so the stream appears in the Streams tab if the SPSS Modeler user interface is visible). As an alternative, to use a specific, unlabeled version:

```
stream = repo.retrieveStream("/projects/retention/risk_score.str", "0:2015-10-12 14:15:41.281", None, True)
```

Note: If both the version and label parameters are `None`, then the latest version is returned.

Storing objects in the repository

To use scripting to store objects in the repository, use the `store*` functions. A summary of the store functions is shown in the following table.

Table 2. Store scripting functions

Object Type	Repository Function
Stream	repo.storeStream(ProcessorStream stream, String path, String label)
Model	repo.storeModel(ModelOutput modelOutput, String path, String label)
Output	repo.storeDocument(DocumentOutput documentOutput, String path, String label)
Node	repo.storeProcessor(Processor node, String path, String label)

For example, you can store a new version of the `risk_score.str` stream with the following function:

```
versionId = repo.storeStream(stream, "/projects/retention/risk_score.str", "test")
```

This example stores a new version of the stream, associates the "test" label with it, and returns the version marker for the newly created version.

Note: If you do not want to associate a label with the new version, pass `None` for the label.

Managing repository folders

By using folders within the repository, you can organize objects into logical groups and make it easier to see which objects are related. Create folders by using the `createFolder()` function, as in the following example:

```
newpath = repo.createFolder("/projects", "cross-sell")
```

This example creates a new folder that is called "cross-sell" in the "/projects" folder. The function returns the full path to the new folder.

To rename a folder, use the `renameFolder()` function:

```
repo.renameFolder("/projects/cross-sell", "cross-sell-Q1")
```

The first parameter is the full path to the folder to be renamed, and the second is the new name to give that folder.

To delete an empty folder, use the `deleteFolder()` function:

```
repo.deleteFolder("/projects/cross-sell")
```

Locking and unlocking objects

From a script, you can lock an object to prevent other users from updating any of its existing versions or creating new versions. You can also unlock an object that you have locked.

The syntax to lock and unlock an object is:

```
repo.lockFile(REPOSITORY_PATH)
repo.lockFile(URI)
```

```
repo.unlockFile(REPOSITORY_PATH)
repo.unlockFile(URI)
```

As with storing and retrieving objects, the `REPOSITORY_PATH` gives the location of the object in the repository. The path must be enclosed in quotation marks and use forward slashes as delimiters. It is not case sensitive.

```
repo.lockFile("/myfolder/Stream1.str")
repo.unlockFile("/myfolder/Stream1.str")
```

Alternatively, you can use a Uniform Resource Identifier (URI) rather than a repository path to give the location of the object. The URI must include the prefix `spsscr:` and must be fully enclosed in quotation marks. Only forward slashes are allowed as path delimiters, and spaces must be encoded. That is, use %20 instead of a space in the path. The URI is not case sensitive. Here are some examples:

```
repo.lockFile("spsscr:///myfolder/Stream1.str")
repo.unlockFile("spsscr:///myfolder/Stream1.str")
```

Note that object locking applies to all versions of an object - you cannot lock or unlock individual versions.

Related information

- [Scripting tips](#)
- [Modifying stream execution](#)
- [Looping through nodes](#)
- [Generating an encoded password](#)
- [Script checking](#)
- [Scripting from the command line](#)
- [Compatibility with previous releases](#)

Generating an encoded password

In certain cases, you may need to include a password in a script; for example, you may want to access a password-protected data source. Encoded passwords can be used in:

- Node properties for Database Source and Output nodes
- Command line arguments for logging into the server
- Database connection properties stored in a .par file (the parameter file generated from the Publish tab of an export node)

Through the user interface, a tool is available to generate encoded passwords based on the Blowfish algorithm (see <http://www.schneier.com/blowfish.html> for more information). Once encoded, you can copy and store the password to script files and command line arguments. The node property **epassword** used for **databasenode** and **databaseexportnode** stores the encoded password.

1. To generate an encoded password, from the Tools menu choose:
 Encode Password...
2. Specify a password in the Password text box.
3. Click Encode to generate a random encoding of your password.
4. Click the Copy button to copy the encoded password to the Clipboard.
5. Paste the password to the desired script or parameter.

Script checking

You can quickly check the syntax of all types of scripts by clicking the red check button on the toolbar of the Standalone Script dialog box.

Figure 1. Stream script toolbar icons



Script checking alerts you to any errors in your code and makes recommendations for improvement. To view the line with errors, click on the feedback in the lower half of the dialog box. This highlights the error in red.

Scripting from the command line

Scripting enables you to run operations typically performed in the user interface. Simply specify and run a standalone script on the command line when launching IBM® SPSS® Modeler. For example:

```
client -script scores.txt -execute
```

The **-script** flag loads the specified script, while the **-execute** flag executes all commands in the script file.

Compatibility with previous releases

Scripts created in previous releases of IBM® SPSS® Modeler should generally work unchanged in the current release. However, model nuggets may now be inserted in the stream automatically (this is the default setting), and may either replace or supplement an existing nugget of that type in the stream. Whether this actually happens depends on the settings of the Add model to stream and Replace previous model options (Tools > Options > User Options > Notifications). You may, for example, need to modify a script from a previous release in which nugget replacement is handled by deleting the existing nugget and inserting the new one.

Scripts created in the current release may not work in earlier releases.

If a script created in an older release uses a command that has since been replaced (or deprecated), the old form will still be supported, but a warning message will be displayed. For example, the old **generated** keyword has been replaced by **model**, and **clear generated** has been replaced by **clear generated palette**. Scripts that use the old forms will still run, but a warning will be displayed.

Accessing stream execution results

Many IBM® SPSS® Modeler nodes produce output objects such as models, charts, and tabular data. Many of these outputs contain useful values that can be used by scripts to guide subsequent runs. These values are grouped into content containers (referred to as simply containers) which

can be accessed using tags or IDs that identify each container. The way these values are accessed depends on the format or "content model" used by that container.

For example, many predictive model outputs use a variant of XML called PMML to represent information about the model such as which fields a decision tree uses at each split, or how the neurones in a neural network are connected and with what strengths. Model outputs that use PMML provide an XML Content Model that can be used to access that information. For example:

```
stream = modeler.script.stream()
# Assume the stream contains a single C5.0 model builder node
# and that the datasource, predictors and targets have already been
# set up
modelbuilder = stream.findByType("c50", None)
results = []
modelbuilder.run(results)
modeloutput = results[0]

# check how many contents are there and what are their names
tags = modeloutput.getContentModelTags()
print "Content Model Tags :" , tags

# Now that we have the C5.0 model output object, access the
# relevant content model
cm = modeloutput.getContentModel("PMML")
if (cm != None) :
    # The PMML content model is a generic XML-based content model that
    # uses XPath syntax. Use that to find the names of the data fields.
    # The call returns a list of strings match the XPath values
    dataFieldNames = cm.getStringValues("/PMML/DataDictionary/DataField", "name")
    print "Data Field Names:" , dataFieldNames
```

IBM SPSS Modeler supports the following content models in scripting:

- Table content model provides access to the simple tabular data represented as rows and columns.
- XML content model provides access to content stored in XML format.
- JSON content model provides access to content stored in JSON format.
- Column statistics content model provides access to summary statistics about a specific field.
- Pair-wise column statistics content model provides access to summary statistics between two fields or values between two separate fields.

Note that the following nodes don't contain these content models:

- Time Series
- Discriminant
- SLRM
- TCM
- All Python nodes
- All Spark nodes
- All Database Modeling nodes
- Extension Model
- STP
- [Table content model](#)
- [XML Content Model](#)
- [JSON Content Model](#)
- [Column statistics content model and pairwise statistics content model](#)

Table content model

The table content model provides a simple model for accessing simple row and column data. The values in a particular column must all have the same type of storage (for example, strings or integers).

API

Table 1. API

Return	Method	Description
int	<code>getRowCount()</code>	Returns the number of rows in this table.
int	<code>getColumnCount()</code>	Returns the number of columns in this table.
String	<code>getColumnName(int columnIndex)</code>	Returns the name of the column at the specified column index. The column index starts at 0.

Return	Method	Description
<code>StorageType</code>	<code>getStorageType(int columnIndex)</code>	Returns the storage type of the column at the specified index. The column index starts at 0.
<code>Object</code>	<code>getValueAt(int rowIndex, int columnIndex)</code>	Returns the value at the specified row and column index. The row and column indices start at 0.
<code>void</code>	<code>reset()</code>	Flushes any internal storage associated with this content model.

Nodes and outputs

This table lists nodes that build outputs which include this type of content model.

Table 2. Nodes and outputs

Node name	Output name	Container ID
<code>table</code>	<code>table</code>	"table"

Example script

```

stream = modeler.script.stream()
from modeler.api import StorageType

# Set up the variable file import node
varfilenode = stream.createAt("variablefile", "DRUG Data", 96, 96)
varfilenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")

# Next create the aggregate node and connect it to the variable file node
aggregatenode = stream.createAt("aggregate", "Aggregate", 192, 96)
stream.link(varfilenode, aggregatenode)

# Configure the aggregate node
aggregatenode.setPropertyValue("keys", ["Drug"])
aggregatenode.setKeyedPropertyValue("aggregates", "Age", ["Min", "Max"])
aggregatenode.setKeyedPropertyValue("aggregates", "Na", ["Mean", "SD"])

# Then create the table output node and connect it to the aggregate node
tablenode = stream.createAt("table", "Table", 288, 96)
stream.link(aggregatenode, tablenode)

# Execute the table node and capture the resulting table output object
results = []
tablenode.run(results)
tableoutput = results[0]

# Access the table output's content model
tablecontent = tableoutput.getContentModel("table")

# For each column, print column name, type and the first row
# of values from the table content
col = 0
while col < tablecontent.getColumnCount():
    print tablecontent.getColumnName(col), \
        tablecontent.getStorageType(col), \
        tablecontent.getValueAt(0, col)
    col = col + 1

```

The output in the scripting Debug tab will look something like this:

```

Age_Min Integer 15
Age_Max Integer 74
Na_Mean Real 0.730851098901
Na_SDev Real 0.116669731242
Drug String drugY
Record_Count Integer 91

```

Related information

- [Accessing stream execution results](#)
- [XML Content Model](#)
- [JSON Content Model](#)
- [Column statistics content model and pairwise statistics content model](#)

XML Content Model

The XML Content Model provides access to XML-based content.

The XML Content Model supports the ability to access components based on XPath expressions. XPath expressions are strings that define which elements or attributes are required by the caller. The XML Content Model hides the details of constructing various objects and compiling expressions that are typically required by XPath support. This makes it simpler to call from Python scripting.

The XML Content Model includes a function that returns the XML document as a string. This allows Python script users to use their preferred Python library to parse the XML.

API

Table 1. API

Return	Method	Description
<code>String</code>	<code>getXMLAsString()</code>	Returns the XML as a string.
<code>number</code>	<code>getNumericValue(String xpath)</code>	Returns the result of evaluating the path with return type of numeric (for example, count the number of elements that match the path expression).
<code>boolean</code>	<code>getBooleanValue(String xpath)</code>	Returns the boolean result of evaluating the specified path expression.
<code>String</code>	<code>getStringValue(String xpath, String attribute)</code>	Returns either the attribute value or XML node value that matches the specified path.
<code>List of strings</code>	<code>getStringValues(String xpath, String attribute)</code>	Returns a list of all attribute values or XML node values that match the specified path.
<code>List of lists of strings</code>	<code>getValuesList(String xpath, <List of strings> attributes, boolean includeValue)</code>	Returns a list of all attribute values that match the specified path along with the XML node value if required.
<code>Hash table (key:string, value:list of string)</code>	<code>getValuesMap(String xpath, String keyAttribute, <List of strings> attributes, boolean includeValue)</code>	Returns a hash table that uses either the key attribute or XML node value as key, and the list of specified attribute values as table values.
<code>boolean</code>	<code>isNamespaceAware()</code>	Returns whether the XML parsers should be aware of namespaces. Default is <code>False</code> .
<code>void</code>	<code>setNamespaceAware(boolean value)</code>	Sets whether the XML parsers should be aware of namespaces. This also calls <code>reset()</code> to ensure changes are picked up by subsequent calls.
<code>void</code>	<code>reset()</code>	Flushes any internal storage associated with this content model (for example, a cached DOM object).

Nodes and outputs

This table lists nodes that build outputs which include this type of content model.

Table 2. Nodes and outputs

Node name	Output name	Container ID
<code>Most model builders</code>	<code>Most generated models</code>	"PMML"
<code>"autodataprep"</code>	n/a	"PMML"

Example script

The Python scripting code to access the content might look like this:

```
results = []
modelbuilder.run(results)
modeloutput = results[0]
cm = modeloutput.getContentModel("PMML")

dataFieldNames = cm.getStringValues("/PMML/DataDictionary/DataField", "name")
predictedNames = cm.getStringValues("//MiningSchema/MiningField[@usageType='predicted']", "name")
```

Related information

- [Accessing stream execution results](#)
- [Table content model](#)
- [JSON Content Model](#)
- [Column statistics content model and pairwise statistics content model](#)

JSON Content Model

The JSON Content Model is used to provide support for JSON format content. This provides a basic API to allow callers to extract values on the assumption that they know which values are to be accessed.

API

Table 1. API

Return	Method	Description
<code>String</code>	<code>getJSONAsString()</code>	Returns the JSON content as a string.
<code>Object</code>	<code>getObjectAt(<List of objects> path, JSONArtifact artifact)</code> throws <code>Exception</code>	Returns the object at the specified path. The supplied root artifact may be null in which case the root of the content is used. The returned value may be a literal string, integer, real or boolean, or a JSON artifact (either a JSON object or a JSON array).
<code>Hash table<(key:object, value:object)></code>	<code>getChildValuesAt(<List of objects> path, JSONArtifact artifact)</code> throws <code>Exception</code>	Returns the child values of the specified path if the path leads to a JSON object or null otherwise. The keys in the table are strings while the associated value may be a literal string, integer, real or boolean, or a JSON artifact (either a JSON object or a JSON array).
<code>List of objects</code>	<code>getChildrenAt(<List of objects> path, JSONArtifact artifact)</code> throws <code>Exception</code>	Returns the list of objects at the specified path if the path leads to a JSON array or null otherwise. The returned values may be a literal string, integer, real or boolean, or a JSON artifact (either a JSON object or a JSON array).
<code>void</code>	<code>reset()</code>	Flushes any internal storage associated with this content model (for example, a cached DOM object).

Nodes and outputs

This table lists nodes that build outputs which include this type of content model.

Table 2. Nodes and outputs

Node name	Output name	Container ID
"applykmeansas"	n/a	"JSON_MV"
"applyxgboosttree"	n/a	"JSON_MV"

Example scripts

The following scripts retrieve JSON files:

```
applykmeansas = stream.findByName("applykmeansas",None)
mvjson = applykmeansas.getContentModel("JSON_MV")
print(mvjson.getJSONAsString())

applyxgboosttree = stream.findByName("applyxgboosttree",None)
mvjson = applyxgboosttree.getContentModel("JSON_MV")
print(mvjson.getJSONAsString())
```

Column statistics content model and pairwise statistics content model

The column statistics content model provides access to statistics that can be computed for each field (univariate statistics). The pairwise statistics content model provides access to statistics that can be computed between pairs of fields or values in a field.

The possible statistics measures are:

- `Count`
- `UniqueCount`
- `ValidCount`
- `Mean`
- `Sum`
- `Min`
- `Max`

- Range
- Variance
- StandardDeviation
- StandardErrorOfMean
- Skewness
- SkewnessStandardError
- Kurtosis
- KurtosisStandardError
- Median
- Mode
- Pearson
- Covariance
- TTest
- FTest

Some values are only appropriate from single column statistics while others are only appropriate for pairwise statistics.

Nodes that will produce these are:

- Statistics node produces column statistics and can produce pairwise statistics when correlation fields are specified
- Data Audit node produces column and can produce pairwise statistics when an overlay field is specified.
- Means node produces pairwise statistics when comparing pairs of fields or comparing a field's values with other field summaries.

Which content models and statistics are available will depend on both the particular node's capabilities and the settings within the node.

ColumnStatsContentModel API

Table 1. ColumnStatsContentModel API

Return	Method	Description
<code>List<StatisticType></code>	<code>getAvailableStatistics()</code>	Returns the available statistics in this model. Not all fields will necessarily have values for all statistics.
<code>List<String></code>	<code>getAvailableColumns()</code>	Returns the column names for which statistics were computed.
<code>Number</code>	<code>getStatistic(String column, StatisticType statistic)</code>	Returns the statistic values associated with the column.
<code>void</code>	<code>reset()</code>	Flushes any internal storage associated with this content model.

PairwiseStatsContentModel API

Table 2. PairwiseStatsContentModel API

Return	Method	Description
<code>List<StatisticType></code>	<code>getAvailableStatistics()</code>	Returns the available statistics in this model. Not all fields will necessarily have values for all statistics.
<code>List<String></code>	<code>getAvailablePrimaryColumns()</code>	Returns the primary column names for which statistics were computed.
<code>List<Object></code>	<code>getAvailablePrimaryValues()</code>	Returns the values of the primary column for which statistics were computed.
<code>List<String></code>	<code>getAvailableSecondaryColumns()</code>	Returns the secondary column names for which statistics were computed.
<code>Number</code>	<code>getStatistic(String primaryColumn, String secondaryColumn, StatisticType statistic)</code>	Returns the statistic values associated with the columns.
<code>Number</code>	<code>getStatistic(String primaryColumn, Object primaryValue, String secondaryColumn, StatisticType statistic)</code>	Returns the statistic values associated with the primary column value and the secondary column.
<code>void</code>	<code>reset()</code>	Flushes any internal storage associated with this content model.

Nodes and outputs

This table lists nodes that build outputs which include this type of content model.

Table 3. Nodes and outputs

Node name	Output name	Container ID	Notes
"means" (Means node)	"means"	"columnStatistics"	
"means" (Means node)	"means"	"pairwiseStatistics"	
"dataaudit" (Data Audit node)	"means"	"columnStatistics"	
"statistics" (Statistics node)	"statistics"	"columnStatistics"	Only generated when specific fields are examined.
"statistics" (Statistics node)	"statistics"	"pairwiseStatistics"	Only generated when fields are correlated.

Example script

```
from modeler.api import StatisticType
stream = modeler.script.stream()

# Set up the input data
varfile = stream.createAt("variablefile", "File", 96, 96)
varfile.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")

# Now create the statistics node. This can produce both
# column statistics and pairwise statistics
statisticsnode = stream.createAt("statistics", "Stats", 192, 96)
statisticsnode.setPropertyValue("examine", ["Age", "Na", "K"])
statisticsnode.setPropertyValue("correlate", ["Age", "Na", "K"])
stream.link(varfile, statisticsnode)

results = []
statisticsnode.run(results)
statsoutput = results[0]
statscm = statsoutput.getContentModel("columnStatistics")
if (statscm != None):
    cols = statscm.getAvailableColumns()
    stats = statscm.getAvailableStatistics()
    print "Column stats:", cols[0], str(stats[0]), " = ", statscm.getStatistic(cols[0], stats[0])

statscm = statsoutput.getContentModel("pairwiseStatistics")
if (statscm != None):
    pcols = statscm.getAvailablePrimaryColumns()
    scols = statscm.getAvailableSecondaryColumns()
    stats = statscm.getAvailableStatistics()
    corr = statscm.getStatistic(pcols[0], scols[0], StatisticType.Pearson)
    print "Pairwise stats:", pcols[0], scols[0], " Pearson = ", corr
```

Command Line Arguments

- [Invoking the software](#)
- [Using command line arguments](#)

Invoking the software

Windows

You can use the command line of your operating system to launch IBM® SPSS® Modeler as follows:

1. On a computer where IBM SPSS Modeler is installed, open a DOS, or command-prompt, window.
2. To launch the IBM SPSS Modeler interface in interactive mode, type the `modelerclient` command followed by the required arguments; for example:

```
modelerclient -stream report.str -execute
```

The available arguments (flags) allow you to connect to a server, load streams, run scripts, or specify other parameters as needed.

Mac OS

1. Locate the Mac OS command path for IBM SPSS Modeler (for example, [Installpath]/IBM SPSS Modeler.app/Contents/MacOS).
2. To launch the IBM SPSS Modeler interface in interactive mode, run the `modeler` command followed by the required arguments; for example:

```
./modeler -stream report.str -execute
```

Using command line arguments

You can append command line arguments (also referred to as *flags*) to the initial `modelerclient` command to alter the invocation of IBM® SPSS® Modeler.

Several types of command line arguments are available, and are described later in this section.

Table 1. Types of command line arguments

Argument type	Where described
System arguments	See the topic System arguments for more information.
Parameter arguments	See the topic Parameter arguments for more information.
Server connection arguments	See the topic Server connection arguments for more information.
IBM SPSS Collaboration and Deployment Services Repository connection arguments	See the topic IBM SPSS Collaboration and Deployment Services Repository Connection Arguments for more information.
IBM SPSS Analytic Server connection arguments	See the topic IBM SPSS Analytic Server connection arguments for more information.

For example, you can use the `-server`, `-stream` and `-execute` flags to connect to a server and then load and run a stream, as follows:

```
modelerclient -server -hostname myserver -port 80 -username dminer  
-password 1234 -stream mystream.str -execute
```

Note that when running against a local client installation, the server connection arguments are not required.

Parameter values that contain spaces can be enclosed in double quotes—for example:

```
modelerclient -stream mystream.str -Pusername="Joe User" -execute
```

You can also execute IBM SPSS Modeler states and scripts in this manner, using the `-state` and `-script` flags, respectively.

Note: If you use a structured parameter in a command, you must precede quotation marks with a backslash. This prevents the quotation marks being removed during interpretation of the string.

Debugging command line arguments

To debug a command line, use the `modelerclient` command to launch IBM SPSS Modeler with the desired arguments. This enables you to verify that commands will execute as expected. You can also confirm the values of any parameters passed from the command line in the Session Parameters dialog box (Tools menu, Set Session Parameters).

- [System arguments](#)
- [Parameter arguments](#)
- [Server connection arguments](#)
- [IBM SPSS Collaboration and Deployment Services Repository Connection Arguments](#)
- [IBM SPSS Analytic Server connection arguments](#)
- [Combining Multiple Arguments](#)

System arguments

The following table describes system arguments available for command line invocation of the user interface.

Table 1. System arguments

Argument	Behavior/Description
@<commandFile>	The @ character followed by a filename specifies a command list. When <code>modelerclient</code> encounters an argument beginning with @, it operates on the commands in that file as if they had been on the command line. See the topic Combining Multiple Arguments for more information.
-directory <dir>	Sets the default working directory. In local mode, this directory is used for both data and output. Example: <code>-directory c:/</code> or <code>-directory c:\\</code>
-server_directory <dir>	Sets the default server directory for data. The working directory, specified by using the <code>-directory</code> flag, is used for output.
-execute	After starting, execute any stream, state, or script loaded at startup. If a script is loaded in addition to a stream or state, the script alone will be executed.

Argument	Behavior/Description
-stream <stream>	At startup, load the stream specified. Multiple streams can be specified, but the last stream specified will be set as the current stream.
-script <script>	At startup, load the standalone script specified. This can be specified in addition to a stream or state as described below, but only one script can be loaded at startup.
-model <model>	At startup, load the generated model (.gm format file) specified.
-state <state>	At startup, load the saved state specified.
-project <project>	Load the specified project. Only one project can be loaded at startup.
-output <output>	At startup, load the saved output object (.cou format file).
-help	Display a list of command line arguments. When this option is specified, all other arguments are ignored and the Help screen is displayed.
-P <name>=<value>	Used to set a startup parameter. Can also be used to set node properties (slot parameters).

Note: Default directories can also be set in the user interface. To access the options, from the File menu, choose Set Working Directory or Set Server Directory.

Loading multiple files

From the command line, you can load multiple streams, states, and outputs at startup by repeating the relevant argument for each object loaded. For example, to load and run two streams called report.str and train.str, you would use the following command:

```
modelerclient -stream report.str -stream train.str -execute
```

Loading objects from the IBM SPSS Collaboration and Deployment Services Repository

Because you can load certain objects from a file or from the IBM® SPSS® Collaboration and Deployment Services Repository (if licensed), the filename prefix **spsscr:** and, optionally, **file:** (for objects on disk) tells IBM SPSS Modeler where to look for the object. The prefix works with the following flags:

- **-stream**
- **-script**
- **-output**
- **-model**
- **-project**

You use the prefix to create a URI that specifies the location of the object—for example, **-stream**

"spsscr://folder_1/scoring_stream.str". The presence of the **spsscr:** prefix requires that a valid connection to the IBM SPSS Collaboration and Deployment Services Repository has been specified in the same command. So, for example, the full command would look like this:

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080  
-spsscr_username myusername -spsscr_password mypassword  
-stream "spsscr://folder_1/scoring_stream.str" -execute
```

Note that from the command line, you *must* use a URI. The simpler **REPOSITORY_PATH** is not supported. (It works only within scripts.) For more details about URIs for objects in the IBM SPSS Collaboration and Deployment Services Repository, see the topic [Accessing Objects in the IBM SPSS Collaboration and Deployment Services Repository](#).

Parameter arguments

Parameters can be used as flags during command line execution of IBM® SPSS® Modeler. In command line arguments, the **-P** flag is used to denote a parameter of the form **-P <name>=<value>**.

Parameters can be any of the following:

- Simple parameters (or parameters used directly in CLEM expressions).
- Slot parameters, also referred to as node properties. These parameters are used to modify the settings of nodes in the stream. See the topic [Node properties overview](#) for more information.
- Command line parameters, used to alter the invocation of IBM SPSS Modeler.

For example, you can supply data source user names and passwords as a command line flag, as follows:

```
modelerclient -stream response.str -P:databasenode.datasource="{"ORA_10gR2",user1,mypsw,false}"
```

The format is the same as that of the **datasource** parameter of the **databasenode** node property. For more information, see: [databasenode properties](#).

The last parameter should be set to **true** if you're passing an encoded password. Also note that no leading spaces should be used in front of the database user name and password (unless, of course, your user name or password actually contains a leading space).

Note: If the node is named, you must surround the node name with double quotes and escape the quotes with a backslash. For example, if the data source node in the preceding example has the name **Source_ABC** the entry would be as follows:

```
modelerclient -stream response.str -P:databasenode.\\"Source_ABC\".datasource="{"ORA_10gR2",user1,mypsw,true}"
```

A backslash is also required in front of the quotes that identify a structured parameter, as in the following TM1 datasource example:

```
clemb -server -hostname 9.115.21.169 -port 28053 -username administrator  
-execute -stream C:\Share\TM1_Script.str -P:tm1import.pm_host="http://9.115.21.163:9510/pmhup/pm"  
-P:tm1import.tm1_connection={"SData","","","admin","\apple"}  
-P:tm1import.selected_view={"SalesPriorCube","\salesmargin%\"}
```

Note: If the database name (in the **datasource** property) contains one or more spaces, periods (also known as a "full stop"), or underscores, you can use the "backslash double quote" format to treat it as string. For example: `"\"db2v9.7.6_linux\""` or: `"\"TADATA_131\""`. In addition, always enclose **datasource** string values in double quotes and curly braces, as in the following example: `"\"SQL Server\",spssuser,abcd1234,false\""`.

Server connection arguments

The **-server** flag tells IBM® SPSS® Modeler that it should connect to a public server, and the flags **-hostname**, **-use_ssl**, **-port**, **-username**, **-password**, and **-domain** are used to tell IBM SPSS Modeler how to connect to the public server. If no **-server** argument is specified, the default or local server is used.

Examples

To connect to a public server:

```
modelerclient -server -hostname myserver -port 80 -username dminer  
-password 1234 -stream mystream.str -execute
```

To connect to a server cluster:

```
modelerclient -server -cluster "QA Machines" \  
-spsscrr_hostname pes_host -spsscrr_port 8080 \  
-spsscrr_username asmith -spsscrr_epassword xyz
```

Note that connecting to a server cluster requires the Coordinator of Processes through IBM SPSS Collaboration and Deployment Services, so the **-cluster** argument must be used in combination with the repository connection options (**spsscrr_***). See the topic [IBM SPSS Collaboration and Deployment Services Repository Connection Arguments](#) for more information.

Table 1. Server connection arguments

Argument	Behavior/Description
-server	Runs IBM SPSS Modeler in server mode, connecting to a public server using the flags -hostname , -port , -username , -password , and -domain .
-hostname <name>	The hostname of the server machine. Available in server mode only.
-use_ssl	Specifies that the connection should use SSL (secure socket layer). This flag is optional; the default setting is <i>not</i> to use SSL.
-port <number>	The port number of the specified server. Available in server mode only.
-cluster <name>	Specifies a connection to a server cluster rather than a named server; this argument is an alternative to the hostname , port and use_ssl arguments. The name is the cluster name, or a unique URI which identifies the cluster in the IBM SPSS Collaboration and Deployment Services Repository. The server cluster is managed by the Coordinator of Processes through IBM SPSS Collaboration and Deployment Services. See the topic IBM SPSS Collaboration and Deployment Services Repository Connection Arguments for more information.
-username <name>	The user name with which to log on to the server. Available in server mode only.
-password <password>	The password with which to log on to the server. Available in server mode only. Note: If the -password argument is not used, you will be prompted for a password.

Argument	Behavior/Description
-epassword <encodedpasswordstring>	The encoded password with which to log on to the server. Available in server mode only. Note: An encoded password can be generated from the Tools menu of the IBM SPSS Modeler application.
-domain <name>	The domain used to log on to the server. Available in server mode only.
-P <name>= <value>	Used to set a startup parameter. Can also be used to set node properties (slot parameters).

IBM SPSS Collaboration and Deployment Services Repository Connection Arguments

If you want to store or retrieve objects from IBM® SPSS® Collaboration and Deployment Services via the command line, you must specify a valid connection to the IBM SPSS Collaboration and Deployment Services Repository. For example:

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080
-spsscr_username myusername -spsscr_password mypassword
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

The following table lists the arguments that can be used to set up the connection.

Table 1. IBM SPSS Collaboration and Deployment Services Repository connection arguments

Argument	Behavior/Description
-spsscr_hostname <hostname or IP address>	The hostname or IP address of the server on which the IBM SPSS Collaboration and Deployment Services Repository is installed.
-spsscr_port <number>	The port number on which the IBM SPSS Collaboration and Deployment Services Repository accepts connections (typically, 8080 by default).
-spsscr_use_ssl	Specifies that the connection should use SSL (secure socket layer). This flag is optional; the default setting is <i>not</i> to use SSL.
-spsscr_username <name>	The user name with which to log on to the IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_password <password>	The password with which to log on to the IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_epassword <encoded password>	The encoded password with which to log on to the IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_providername <name>	The authentication provider used for logging on to the IBM SPSS Collaboration and Deployment Services Repository (Active Directory or LDAP). This is not required if using the native (Local Repository) provider.

For more information about storing and retrieving objects from the IBM SPSS Collaboration and Deployment Services Repository via the command line, see [System arguments](#).

IBM SPSS Analytic Server connection arguments

If you want to store or retrieve objects from IBM® SPSS® Analytic Server via the command line, you must specify a valid connection to IBM SPSS Analytic Server.

Note: The default location of Analytic Server is obtained from SPSS Modeler Server. Users can also define their own Analytic Server connections via Tools > Analytic Server Connections.

The following table lists the arguments that can be used to set up the connection.

Table 1. IBM SPSS Analytic Server connection arguments

Argument	Behavior/Description
-analytic_server_username	The user name with which to log on to IBM SPSS Analytic Server.
-analytic_server_password	The password with which to log on to IBM SPSS Analytic Server.
-analytic_server_epassword	The encoded password with which to log on to IBM SPSS Analytic Server.
-analytic_server_credential	The credentials used to log on to IBM SPSS Analytic Server.

Combining Multiple Arguments

Multiple arguments can be combined in a single command file specified at invocation by using the @ symbol followed by the filename. This enables you to shorten the command line invocation and overcome any operating system limitations on command length. For example, the following startup command uses the arguments specified in the file referenced by <commandFileName>.

```
modelerclient @<commandFileName>
```

Enclose the filename and path to the command file in quotation marks if spaces are required, as follows:

```
modelerclient @ "C:\Program Files\IBM\SPSS\Modeler\nn\scripts\my_command_file.txt"
```

The command file can contain all arguments previously specified individually at startup, with one argument per line. For example:

```
-stream report.str  
-Porder.full_filename=APR_orders.dat  
-Report.filename=APR_report.txt  
-execute
```

When writing and referencing command files, be sure to follow these constraints:

- Use only one command per line.
- Do not embed an @CommandFile argument within a command file.

Properties Reference

- [Properties reference overview](#)
- [Node properties overview](#)

Properties reference overview

You can specify a number of different properties for nodes, streams, projects, and SuperNodes. Some properties are common to all nodes, such as name, annotation, and ToolTip, while others are specific to certain types of nodes. Other properties refer to high-level stream operations, such as caching or SuperNode behavior. Properties can be accessed through the standard user interface (for example, when you open a dialog box to edit options for a node) and can also be used in a number of other ways.

- Properties can be modified through scripts, as described in this section. For more information, see [Syntax for properties](#).
- Node properties can be used in SuperNode parameters.
- Node properties can also be used as part of a command line option (using the -P flag) when starting IBM® SPSS® Modeler.

In the context of scripting within IBM SPSS Modeler, node and stream properties are often called **slot parameters**. In this guide, they are referred to as node or stream properties.

- [Syntax for properties](#)
- [Node and stream property examples](#)

Syntax for properties

Properties can be set using the following syntax

```
OBJECT.setPropertyValue(PROPERTY, VALUE)
```

or:

```
OBJECT.setKeyedPropertyValue(PROPERTY, KEY, VALUE)
```

The value of properties can be retrieved using the following syntax:

```
VARIABLE = OBJECT.getPropertyValue(PROPERTY)
```

or:

```
VARIABLE = OBJECT.getKeyedPropertyValue(PROPERTY, KEY)
```

where **OBJECT** is a node or output, **PROPERTY** is the name of the node property that your expression refers to, and **KEY** is the key value for keyed properties.. For example, the following syntax is used to find the filter node, and then set the default to include all fields and filter the **Age** field from downstream data:

```

filternode = modeler.script.stream().findByType("filter", None)
filternode.setPropertyValue("default_include", True)
filternode.setKeyedPropertyValue("include", "Age", False)

```

All nodes used in IBM® SPSS® Modeler can be located using the stream `findByType (TYPE, LABEL)` function. At least one of `TYPE` or `LABEL` must be specified.

- [Structured properties](#)
- [Abbreviations](#)

Structured properties

There are two ways in which scripting uses structured properties for increased clarity when parsing:

- To give structure to the names of properties for complex nodes, such as Type, Filter, or Balance nodes.
- To provide a format for specifying multiple properties at once.

Structuring for Complex Interfaces

The scripts for nodes with tables and other complex interfaces (for example, the Type, Filter, and Balance nodes) must follow a particular structure in order to parse correctly. These properties need a name that is more complex than the name for a single identifier, this name is called the key. For example, within a Filter node, each available field (on its upstream side) is switched on or off. In order to refer to this information, the Filter node stores one item of information per field (whether each field is true or false). This property may have (or be given) the value `True` or `False`. Suppose that a Filter node named `mynode` has (on its upstream side) a field called `Age`. To switch this to off, set the property `include`, with the key `Age`, to the value `False`, as follows:

```

mynode.setKeyedPropertyValue("include", "Age", False)

```

Structuring to Set Multiple Properties

For many nodes, you can assign more than one node or stream property at a time. This is referred to as the **multiset command** or **set block**.

In some cases, a structured property can be quite complex. An example is as follows:

```

sortnode.setPropertyValue("keys", [[{"K": "Descending"}, {"Age": "Ascending"}, {"Na": "Descending"}]])

```

Another advantage that structured properties have is their ability to set several properties on a node before the node is stable. By default, a multiset sets all properties in the block before taking any action based on an individual property setting. For example, when defining a Fixed File node, using two steps to set field properties would result in errors because the node is not consistent until both settings are valid. Defining properties as a multiset circumvents this problem by setting both properties before updating the data model.

Abbreviations

Standard abbreviations are used throughout the syntax for node properties. Learning the abbreviations is helpful in constructing scripts.

Table 1. Standard abbreviations used
throughout the syntax

Abbreviation	Meaning
abs	Absolute value
len	Length
min	Minimum
max	Maximum
correl	Correlation
covar	Covariance
num	Number or numeric
pct	Percent or percentage
transp	Transparency
xval	Cross-validation
var	Variance or variable (in source nodes)

Node and stream property examples

Node and stream properties can be used in a variety of ways with IBM® SPSS® Modeler. They are most commonly used as part of a script, either a **standalone script**, used to automate multiple streams or operations, or a **stream script**, used to automate processes within a single stream. You can also specify node parameters by using the node properties within the SuperNode. At the most basic level, properties can also be used as a command line option for starting IBM SPSS Modeler. Using the **-p** argument as part of command line invocation, you can use a stream property to change a setting in the stream.

Table 1. Node and stream property examples

Property	Meaning
<code>s.max_size</code>	Refers to the property <code>max_size</code> of the node named <code>s</code> .
<code>s:samplenode.max_size</code>	Refers to the property <code>max_size</code> of the node named <code>s</code> , which must be a Sample node.
<code>:samplenode.max_size</code>	Refers to the property <code>max_size</code> of the Sample node in the current stream (there must be only one Sample node).
<code>s:sample.max_size</code>	Refers to the property <code>max_size</code> of the node named <code>s</code> , which must be a Sample node.
<code>t.direction.Age</code>	Refers to the role of the field <code>Age</code> in the Type node <code>t</code> .
<code>:.max_size</code>	*** NOT LEGAL *** You must specify either the node name or the node type.

The example `s:sample.max_size` illustrates that you do not need to spell out node types in full.

The example `t.direction.Age` illustrates that some slot names can themselves be structured—in cases where the attributes of a node are more complex than simply individual slots with individual values. Such slots are called **structured** or **complex** properties.

Node properties overview

Each type of node has its own set of legal properties, and each property has a type. This type may be a general type—number, flag, or string—in which case settings for the property are coerced to the correct type. An error is raised if they cannot be coerced. Alternatively, the property reference may specify the range of legal values, such as `Discard`, `PairAndDiscard`, and `IncludeAsText`, in which case an error is raised if any other value is used. Flag properties should be read or set by using values of `true` and `false`. (Variations including `Off`, `OFF`, `off`, `No`, `NO`, `no`, `n`, `N`, `f`, `F`, `false`, `False`, `FALSE`, or `0` are also recognized when setting values but may cause errors when reading property values in some cases. All other values are regarded as true. Using `true` and `false` consistently will avoid any confusion.) In this guide's reference tables, the structured properties are indicated as such in the Property description column, and their usage formats are given.

- [Common Node Properties](#)

Common Node Properties

A number of properties are common to all nodes (including SuperNodes) in IBM® SPSS® Modeler.

Table 1. Common node properties

Property name	Data type	Property description
<code>use_custom_name</code>	<code>flag</code>	
<code>name</code>	<code>string</code>	Read-only property that reads the name (either auto or custom) for a node on the canvas.
<code>custom_name</code>	<code>string</code>	Specifies a custom name for the node.
<code>tooltip</code>	<code>string</code>	
<code>annotation</code>	<code>string</code>	
<code>keywords</code>	<code>string</code>	Structured slot that specifies a list of keywords associated with the object (for example, <code>["Keyword1", "Keyword2"]</code>).
<code>cache_enabled</code>	<code>flag</code>	
<code>node_type</code>	<code>source_supernode</code> <code>process_supernode</code> <code>terminal_supernode</code> all node names as specified for scripting	Read-only property used to refer to a node by type. For example, instead of referring to a node only by name, such as <code>real_income</code> , you can also specify the type, such as <code>userinputnode</code> or <code>filternode</code> .

Related information

- [Node properties overview](#)

Stream properties

A variety of stream properties can be controlled by scripting. To reference stream properties, you must set the execution method to use scripts:

```
stream = modeler.script.stream()
stream.setPropertyValue("execute_method", "Script")
```

Example

The node property is used to refer to the nodes in the current stream. The following stream script provides an example:

```
stream = modeler.script.stream()
annotation = stream.getPropertyValue("annotation")

annotation = annotation + "\n\nThis stream is called \"" + stream.getLabel() + "\" and
contains the following nodes:\n"

for node in stream.iterator():
    annotation = annotation + "\n" + node.getTypeName() + " node called \"" + node.getLabel()
    + "\""

stream.setPropertyValue("annotation", annotation)
```

The above example uses the node property to create a list of all nodes in the stream and write that list in the stream annotations. The annotation produced looks like this:

This stream is called "druglearn" and contains the following nodes:

```
type node called "Define Types"
derive node called "Na_to_K"
variablefile node called "DRUGin"
neuralnetwork node called "Drug"
c50 node called "Drug"
filter node called "Discard Fields"
```

Stream properties are described in the following table.

Table 1. Stream properties

Property name	Data type	Property description
execute_method	Normal Script	

Property name	Data type	Property description
date_baseline	number	
date_2digit_baseline	number	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H) H: (M) M: (S) S" "(H) H: (M) M" (M) M: (S) S" "HH.MM.SS" "HH.MM" "MM.SS" "(H) H. (M) M. (S) S" "(H) H. (M) M" "(M) M. (S) S"	
time_rollover	flag	
import_datetime_as_string	flag	
decimal_places	number	
decimal_symbol	Default Period Comma	
angles_in_radians	flag	
use_max_set_size	flag	
max_set_size	number	
ruleset_evaluation	Voting FirstHit	
refresh_source_nodes	flag	Use to refresh source nodes automatically upon stream execution.
script	string	
annotation	string	
name	string	Note: This property is read-only. If you want to change the name of a stream, you should save it with a different name.
parameters		Use this property to update stream parameters from within a stand-alone script.
nodes		See detailed information below.
encoding	SystemDefault "UTF-8"	
stream_rewriting	boolean	
stream_rewriting_maximize_sql	boolean	
stream_rewriting_optimize_clem_execution	boolean	
stream_rewriting_optimize_syntax_execution	boolean	
enable_parallelism	boolean	
sql_generation	boolean	
database_caching	boolean	
sql_logging	boolean	
sql_generation_logging	boolean	
sql_log_native	boolean	
sql_log_prettyprint	boolean	
record_count_suppress_input	boolean	
record_count_feedback_interval	integer	
use_stream_auto_create_node_settings	boolean	If true, then stream-specific settings are used, otherwise user preferences are used.
create_model_applier_for_new_models	boolean	If true, when a model builder creates a new model, and it has no active update links, a new model applier is added. Note: If you are using IBM® SPSS® Modeler Batch version 15 you must explicitly add the model applier within your script.

Property name	Data type	Property description
<code>create_model_applier_up_date_links</code>	<code>createEnabled</code> <code>createDisabled</code> <code>doNotCreate</code>	Defines the type of link created when a model applier node is added automatically.
<code>create_source_node_from_builders</code>	<code>boolean</code>	If true, when a source builder creates a new source output, and it has no active update links, a new source node is added.
<code>create_source_node_update_links</code>	<code>createEnabled</code> <code>createDisabled</code> <code>doNotCreate</code>	Defines the type of link created when a source node is added automatically.
<code>has_coordinate_system</code>	<code>boolean</code>	If true, applies a coordinate system to the entire stream.
<code>coordinate_system</code>	<code>string</code>	The name of the selected projected coordinate system.
<code>deployment_area</code>	<code>ModelRefresh</code> <code>Scoring</code> <code>None</code>	Choose how you want to deploy the stream. If this value is set to <code>None</code> , no other deployment entries are used.
<code>scoring_terminal_node_id</code>	<code>string</code>	Choose the scoring branch in the stream. It can be any terminal node in the stream.
<code>scoring_node_id</code>	<code>string</code>	Choose the nugget in the scoring branch.
<code>model_build_node_id</code>	<code>string</code>	Choose the modeling node in the stream.

Source Node Properties

- [Source node common properties](#)
- [asimport Properties](#)
- [cognosimport Node Properties](#)
- [databasenode properties](#)
- [datacollectionimportnode Properties](#)
- [excelimportnode Properties](#)
- [extensionimportnode.properties](#)
- [fixedfilenode Properties](#)
- [gsdata_import Node Properties](#)
- [jsonimportnode Properties](#)
- [sasimportnode Properties](#)
- [simgennode properties](#)
- [statisticimportnode Properties](#)
- [tm1odataimport Node Properties](#)
- [tm1import Node Properties \(deprecated\)](#)
- [twcimport node properties](#)
- [userinputnode properties](#)
- [variablefilenode Properties](#)
- [xmlimportnode Properties](#)

Source node common properties

Properties that are common to all source nodes are listed below, with information on specific nodes in the topics that follow.

Example 1

```
varfilenode = modeler.script.stream().create("variablefile", "Var. File")
varfilenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
varfilenode.setKeyedPropertyValue("check", "Age", "None")
varfilenode.setKeyedPropertyValue("values", "Age", [1, 100])
varfilenode.setKeyedPropertyValue("type", "Age", "Range")
varfilenode.setKeyedPropertyValue("direction", "Age", "Input")
```

Example 2

This script assumes that the specified data file contains a field called `Region` that represents a multi-line string.

```
from modeler.api import StorageType
from modeler.api import MeasureType

# Create a Variable File node that reads the data set containing
# the "Region" field
varfilenode = modeler.script.stream().create("variablefile", "My Geo Data")
```

```

varfilenode.setPropertyValue("full_filename", "C:/mydata/mygeodata.csv")
varfilenode.setPropertyValue("treat_square_brackets_as_lists", True)

# Override the storage type to be a list...
varfilenode.setKeyedPropertyValue("custom_storage_type", "Region", StorageType.LIST)
# ...and specify the type if values in the list and the list depth
varfilenode.setKeyedPropertyValue("custom_list_storage_type", "Region", StorageType.INTEGER)
varfilenode.setKeyedPropertyValue("custom_list_depth", "Region", 2)

# Now change the measurement to indentify the field as a geospatial value...
varfilenode.setKeyedPropertyValue("measure_type", "Region", MeasureType.GEOSPATIAL)
# ...and finally specify the necessary information about the specific
# type of geospatial object
varfilenode.setKeyedPropertyValue("geo_type", "Region", "MultiLineString")
varfilenode.setKeyedPropertyValue("geo_coordinates", "Region", "2D")
varfilenode.setKeyedPropertyValue("has_coordinate_system", "Region", True)
varfilenode.setKeyedPropertyValue("coordinate_system", "Region",
                                "ETRS_1989_EPSG_Arctic_zone_5-47")

```

Table 1. Source node common properties

Property name	Data type	Property description
direction	Input Target Both None Partition Split Frequency RecordID	Keyed property for field roles. Usage format: NODE.direction.FIELDNAME Note: The values In and Out are now deprecated. Support for them may be withdrawn in a future release.
type	Range Flag Set Typeless Discrete Ordered Set Default	Type of field. Setting this property to Default will clear any values property setting, and if value_mode is set to Specify, it will be reset to Read. If value_mode is already set to Pass or Read, it will be unaffected by the type setting. Usage format: NODE.type.FIELDNAME
storage	Unknown String Integer Real Time Date Timestamp	Read-only keyed property for field storage type. Usage format: NODE.storage.FIELDNAME
check	None Nullify Coerce Discard Warn Abort	Keyed property for field type and range checking. Usage format: NODE.check.FIELDNAME
values	[value value]	For a continuous (range) field, the first value is the minimum, and the last value is the maximum. For nominal (set) fields, specify all values. For flag fields, the first value represents false, and the last value represents true. Setting this property automatically sets the value_mode property to Specify. The storage is determined based on the first value in the list, for example, if the first value is a string then the storage is set to String. Usage format: NODE.values.FIELDNAME
value_mode	Read Pass Read+ Current Specify	Determines how values are set for a field on the next data pass. Usage format: NODE.value_mode.FIELDNAME Note that you cannot set this property to Specify directly; to use specific values, set the values property.
default_value_mode	Read Pass	Specifies the default method for setting values for all fields. Usage format: NODE.default_value_mode This setting can be overridden for specific fields by using the value_mode property.
extend_values	flag	Applies when value_mode is set to Read. Set to T to add newly read values to any existing values for the field. Set to F to discard existing values in favor of the newly read values. Usage format: NODE.extend_values.FIELDNAME
value_labels	string	Used to specify a value label. Note that values must be specified first.
enable_missing	flag	When set to T, activates tracking of missing values for the field. Usage format: NODE.enable_missing.FIELDNAME
missing_values	[value value ...]	Specifies data values that denote missing data. Usage format: NODE.missing_values.FIELDNAME
range_missing	flag	When this property is set to T, specifies whether a missing-value (blank) range is defined for a field. Usage format: NODE.range_missing.FIELDNAME
missing_lower	string	When range_missing is true, specifies the lower bound of the missing-value range. Usage format: NODE.missing_lower.FIELDNAME
missing_upper	string	When range_missing is true, specifies the upper bound of the missing-value range. Usage format: NODE.missing_upper.FIELDNAME
null_missing	flag	When this property is set to T, nulls (undefined values that are displayed as \$null\$ in the software) are considered missing values. Usage format: NODE.null_missing.FIELDNAME
whitespace_missing	flag	When this property is set to T, values containing only white space (spaces, tabs, and new lines) are considered missing values. Usage format: NODE.whitespace_missing.FIELDNAME
description	string	Used to specify a field label or description.

Property name	Data type	Property description
<code>default_incl ude</code>	<code>flag</code>	Keyed property to specify whether the default behavior is to pass or filter fields: <code>NODE.default_include Example: set mynode:filternode.default_include = false</code>
<code>include</code>	<code>flag</code>	Keyed property used to determine whether individual fields are included or filtered: <code>NODE.include.FIELDNAME.</code>
<code>new_name</code>	<code>string</code>	
<code>measure_type</code>	<code>Range / MeasureType. RANGE Discrete / MeasureType. DISCRETE Flag / MeasureType. FLAG Set / MeasureType. SET OrderedSet / MeasureType. ORDERED_SET Typeless / MeasureType. TYPELESS Collection / MeasureType. COLLECTION Geospatial / MeasureType. GEOSPATIAL</code>	This keyed property is similar to <code>type</code> in that it can be used to define the measurement associated with the field. What is different is that in Python scripting, the setter function can also be passed one of the <code>MeasureType</code> values while the getter will always return on the <code>MeasureType</code> values.
<code>collection_m easure</code>	<code>Range / MeasureType. RANGE Flag / MeasureType. FLAG Set / MeasureType. SET OrderedSet / MeasureType. ORDERED_SET Typeless / MeasureType. TYPELESS</code>	For collection fields (lists with a depth of 0), this keyed property defines the measurement type associated with the underlying values.
<code>geo_type</code>	<code>Point MultiPoint LineString MultiLineStr ing Polygon MultiPolygon</code>	For geospatial fields, this keyed property defines the type of geospatial object represented by this field. This should be consistent with the list depth of the values.
<code>has_coordinate_system</code>	<code>boolean</code>	For geospatial fields, this property defines whether this field has a coordinate system
<code>coordinate_s ystem</code>	<code>string</code>	For geospatial fields, this keyed property defines the coordinate system for this field.

Property name	Data type	Property description
custom_storage_type	Unknown / MeasureType. UNKNOWN String / MeasureType. STRING Integer / MeasureType. INTEGER Real / MeasureType. REAL Time / MeasureType. TIME Date / MeasureType. DATE Timestamp / MeasureType. TIMESTAMP List / MeasureType. LIST	This keyed property is similar to <code>custom_storage</code> in that it can be used to define the override storage for the field. What is different is that in Python scripting, the setter function can also be passed one of the <code>StorageType</code> values while the getter will always return on the <code>StorageType</code> values.
custom_list_storage_type	String / MeasureType. STRING Integer / MeasureType. INTEGER Real / MeasureType. REAL Time / MeasureType. TIME Date / MeasureType. DATE Timestamp / MeasureType. TIMESTAMP	For list fields, this keyed property specifies the storage type of the underlying values.
custom_list_depth	integer	For list fields, this keyed property specifies the depth of the field
max_list_length	integer	Only available for data with a measurement level of either <i>Geospatial</i> or <i>Collection</i> . Set the maximum length of the list by specifying the number of elements the list can contain.
max_string_length	integer	Only available for <i>typeless</i> data and used when you are generating SQL to create a table. Enter the value of the largest string in your data; this generates a column in the table that is big enough to contain the string.

asimport Properties

The Analytic Server source enables you to run a stream on Hadoop Distributed File System (HDFS).

Example

```
node.setPropertyValue("use_default_as", False)
node.setPropertyValue("connection",
["false","9.119.141.141","9080","analyticserver","ibm","admin","admin","false","","","",""])

```

Table 1. asimport properties

asimport properties	Data type	Property description
data_source	string	The name of the data source.
use_default_as	boolean	If set to <code>True</code> , uses the default Analytic Server connection configured in the server options.cfg file. If set to <code>False</code> , uses the connection of this node.

asimport properties	Data type	Property description
connection	["string", "string", "string", "string", "string", "string", "string", "string"]	A list property containing the Analytic Server connection details. The format is: ["is_secure_connect", "server_url", "server_port", "context_root", "consumer", "user_name", "password", "use-kerberos-auth", "kerberos-krb5-config-file-path", "kerberos-jaas-config-file-path", "kerberos-krb5-service-principal-name", "enable-kerberos-debug"] Where: is_secure_connect: indicates whether secure connection is used, and is either true or false. use-kerberos-auth: indicates whether kerberos authentication is used, and is either true or false. enable-kerberos-debug: indicates whether the debug mode of kerberos authentication is used, and is either true or false.

cognosimport Node Properties



The IBM Cognos source node imports data from Cognos Analytics databases.

Example

```
node = stream.create("cognosimport", "My node")
node.setPropertyValue("cognos_connection", ["http://mycogsrv1:9300/p2pd/servlet/dispatch",
    True, "", "", ""])
node.setPropertyValue("cognos_package_name", "/Public Folders/GOSALES")
node.setPropertyValue("cognos_items", "[GreatOutdoors].[BRANCH].[BRANCH_CODE]", "[GreatOutdoors].[BRANCH].[COUNTRY_CODE]"])
```

Table 1. cognosimport node properties

cognosimport node properties	Data type	Property description
mode	Data Report	Specifies whether to import Cognos data (default) or reports.
cognos_connection	["string", "flag", "string", "string", "string"]	A list property containing the connection details for the Cognos server. The format is: ["Cognos_server_URL", login_mode, namespace, username, password] where: Cognos_server_URL is the URL of the Cognos server containing the source. login_mode indicates whether anonymous login is used, and is either true or false; if set to true, the following fields should be set to "". namespace specifies the security authentication provider used to log on to the server. username and password are those used to log on to the Cognos server. Instead of login_mode, the following modes are also available: <ul style="list-style-type: none"> anonymousMode. For example: ['Cognos_server_url', 'anonymousMode', 'namespace', 'username', 'password'] credentialMode. For example: ['Cognos_server_url', 'credentialMode', 'namespace', 'username', 'password'] storedCredentialMode. For example: ['Cognos_server_url', 'storedCredentialMode', 'stored_credential_name'] Where stored_credential_name is the name of a Cognos credential in the repository.
cognos_package_name	string	The path and name of the Cognos package from which you are importing data objects, for example: /Public Folders/GOSALES Note: Only forward slashes are valid.
cognos_items	["field", "field", ..., "field"]	The name of one or more data objects to be imported. The format of field is [namespace].[query_subject].[query_item]
cognos_filters	field	The name of one or more filters to apply before importing data.
cognos_data_parameters	list	Values for prompt parameters for data. Name-and-value pairs are enclosed in square brackets, and multiple pairs are separated by commas and the whole string enclosed in square brackets. Format: [{"param1", "value"}, ..., {"paramN", "value"}]]
cognos_report_directory	field	The Cognos path of a folder or package from which to import reports, for example: /Public Folders/GOSALES Note: Only forward slashes are valid.
cognos_report_name	field	The path and name within the report location of a report to import.
cognos_report_parameters	list	Values for report parameters. Name-and-value pairs are enclosed in square brackets, and multiple pairs are separated by commas and the whole string enclosed in square brackets. Format: [{"param1", "value"}, ..., {"paramN", "value"}]]

databasenode properties



You can use the Database node to import data from a variety of other packages using ODBC (Open Database Connectivity), including Microsoft SQL Server, Db2, Oracle, and others.

Example

```
import modeler.api
stream = modeler.script.stream()
node = stream.create("database", "My node")
node.setPropertyValue("mode", "Table")
node.setPropertyValue("query", "SELECT * FROM drug1n")
node.setPropertyValue("datasource", "Drug1n_db")
node.setPropertyValue("username", "spss")
node.setPropertyValue("password", "spss")
node.setPropertyValue("tablename", ".Drug1n")
```

Table 1. databasenode properties

databasenode properties	Data type	Property description
mode	Table Query	Specify <i>Table</i> to connect to a database table by using dialog box controls, or specify <i>Query</i> to query the selected database by using SQL.
datasource	string	Database name (see also note below).
username	string	Database connection details (see also note below).
password	string	
credential	string	Name of credential stored in IBM® SPSS® Collaboration and Deployment Services. This can be used instead of the username and password properties. The credential's user name and password must match the user name and password required to access the database
use_credential		Set to True or False .
epassword	string	Specifies an encoded password as an alternative to hard-coding a password in a script. See the topic Generating an encoded password for more information. This property is read-only during execution.
tablename	string	Name of the table you want to access.
strip_spaces	None Left Right Both	Options for discarding leading and trailing spaces in strings.
use_quotes	AsNeeded Always Never	Specify whether table and column names are enclosed in quotation marks when queries are sent to the database (for example, if they contain spaces or punctuation).
query	string	Specifies the SQL code for the query you want to submit.

Note: If the database name (in the **datasource** property) contains spaces, then instead of individual properties for **datasource**, **username** and **password**, you can also use a single **datasource** property in the following format:

Table 2. databasenode properties - datasource specific

databasenode properties	Data type	Property description
datasource	string	Format: [database_name , username , password [, true false]] The last parameter is for use with encrypted passwords. If this is set to true , the password will be decrypted before use.

Use this format also if you are changing the data source; however, if you just want to change the **username** or **password**, you can use the **username** or **password** properties.

datacollectionimportnode Properties



The Data Collection Data Import node imports survey data based on the Data Collection Data Model used by market research products. The Data Collection Data Library must be installed to use this node.

Example

```
node = stream.create("datacollectionimport", "My node")
node.setPropertyValue("metadata_name", "mrQvDsc")
node.setPropertyValue("metadata_file", "C:/Program Files/IBM/SPSS/DataCollection/DDL/Data/Quanvert/Museum/museum.pkd")
node.setPropertyValue("casedata_name", "mrQvDsc")
node.setPropertyValue("casedata_source_type", "File")
node.setPropertyValue("casedata_file", "C:/Program Files/IBM/SPSS/DataCollection/DDL/Data/Quanvert/Museum/museum.pkd")
node.setPropertyValue("import_system_variables", "Common")
node.setPropertyValue("import_multi_response", "MultipleFlags")
```

Table 1. datacollectionimportnode properties

datacollectionimportnode properties	Data type	Property description
<code>metadata_name</code>	<code>string</code>	The name of the MDSC. The special value <code>DimensionsMDD</code> indicates that the standard Data Collection metadata document should be used. Other possible values include: <code>mrADODsc mrI2dDsc mrLogDsc mrQdiDrsDsc mrQvDsc mrSampleReportingMDSC mrSavDsc mrSCDsc mrScriptMDSC</code> The special value <code>none</code> indicates that there is no MDSC.
<code>metadata_file</code>	<code>string</code>	Name of the file where the metadata is stored.
<code>casedata_name</code>	<code>string</code>	The name of the CDSC. Possible values include: <code>mrADODsc mrI2dDsc mrLogDsc mrPunchDSC mrQdiDrsDsc mrQvDsc mrRdbDsc2 mrSavDsc mrScDSC mrXmlDsc</code> The special value <code>none</code> indicates that there is no CDSC.
<code>casedata_source_type</code>	<code>Unknown File Folder UDL DSN</code>	Indicates the source type of the CDSC.
<code>casedata_file</code>	<code>string</code>	When <code>casedata_source_type</code> is <code>File</code> , specifies the file containing the case data.
<code>casedata_folder</code>	<code>string</code>	When <code>casedata_source_type</code> is <code>Folder</code> , specifies the folder containing the case data.
<code>casedata_udl_string</code>	<code>string</code>	When <code>casedata_source_type</code> is <code>UDL</code> , specifies the OLD-DB connection string for the data source containing the case data.
<code>casedata_dsn_string</code>	<code>string</code>	When <code>casedata_source_type</code> is <code>DSN</code> , specifies the ODBC connection string for the data source.
<code>casedata_project</code>	<code>string</code>	When reading case data from a Data Collection database, you can enter the name of the project. For all other case data types, this setting should be left blank.
<code>version_import_mode</code>	<code>All Latest Specify</code>	Defines how versions should be handled.
<code>specific_version</code>	<code>string</code>	When <code>version_import_mode</code> is <code>Specify</code> , defines the version of the case data to be imported.
<code>use_language</code>	<code>string</code>	Defines whether labels of a specific language should be used.
<code>language</code>	<code>string</code>	If <code>use_language</code> is true, defines the language code to use on import. The language code should be one of those available in the case data.
<code>use_context</code>	<code>string</code>	Defines whether a specific context should be imported. Contexts are used to vary the description associated with responses.
<code>context</code>	<code>string</code>	If <code>use_context</code> is true, defines the context to import. The context should be one of those available in the case data.
<code>use_label_type</code>	<code>string</code>	Defines whether a specific type of label should be imported.
<code>label_type</code>	<code>string</code>	If <code>use_label_type</code> is true, defines the label type to import. The label type should be one of those available in the case data.
<code>user_id</code>	<code>string</code>	For databases requiring an explicit login, you can provide a user ID and password to access the data source.
<code>password</code>	<code>string</code>	
<code>import_system_variables</code>	<code>Common None All</code>	Specifies which system variables are imported.
<code>import_codes_variables</code>	<code>flag</code>	
<code>import_sourcefile_variables</code>	<code>flag</code>	
<code>import_multiresponse</code>	<code>MultipleFlags Single</code>	

Related information

- [Node properties overview](#)
- [Source node common properties](#)
- [cognosimport Node Properties](#)
- [databasenode properties](#)
- [excelimportnode Properties](#)
- [evimportnode Properties](#)
- [fixedfilenode Properties](#)
- [sasimportnode Properties](#)
- [simgennode properties](#)
- [userinputnode properties](#)
- [variablefilenode Properties](#)

- [xmlimportnode Properties](#)
- [statisticsimportnode Properties](#)

excelimportnode Properties



The Excel Import node imports data from Microsoft Excel in the .xlsx file format. An ODBC data source is not required.

Examples

```
#To use a named range:  
node = stream.create("excelimport", "My node")  
node.setPropertyValue("excel_file_type", "Excel2007")  
node.setPropertyValue("full_filename", "C:/drug.xlsx")  
node.setPropertyValue("use_named_range", True)  
node.setPropertyValue("named_range", "DRUG")  
node.setPropertyValue("read_field_names", True)  
  
#To use an explicit range:  
node = stream.create("excelimport", "My node")  
node.setPropertyValue("excel_file_type", "Excel2007")  
node.setPropertyValue("full_filename", "C:/drug.xlsx")  
node.setPropertyValue("worksheet_mode", "Name")  
node.setPropertyValue("worksheet_name", "Drug")  
node.setPropertyValue("explicit_range_start", "A1")  
node.setPropertyValue("explicit_range_end", "F300")
```

Table 1. excelimportnode properties

excelimportnode properties	Data type	Property description
excel_file_type	Excel2007	
full_filename	string	The complete filename, including path.
use_named_range	Boolean	Whether to use a named range. If true, the named_range property is used to specify the range to read, and other worksheet and data range settings are ignored.
named_range	string	
worksheet_mode	Index Name	Specifies whether the worksheet is defined by index or name.
worksheet_index	integer	Index of the worksheet to be read, beginning with 0 for the first worksheet, 1 for the second, and so on.
worksheet_name	string	Name of the worksheet to be read.
data_range_mode	FirstNonBlank ExplicitRange	Specifies how the range should be determined.
blank_rows	StopReading ReturnBlankRows	When data_range_mode is <i>FirstNonBlank</i> , specifies how blank rows should be treated.
explicit_range_start	string	When data_range_mode is <i>ExplicitRange</i> , specifies the starting point of the range to read.
explicit_range_end	string	
read_field_names	Boolean	Specifies whether the first row in the specified range should be used as field (column) names.
scanLineCount	integer	Specifies the number of rows to scan for the column and storage type. Default is 200.

extensionimportnode properties



With the Extension Import node, you can run R or Python for Spark scripts to import data.

Python for Spark example

```
##### Script example for Python for Spark  
import modeler.api  
stream = modeler.script.stream()  
node = stream.create("extension_importer", "extension_importer")  
node.setPropertyValue("syntax_type", "Python")
```

```

python_script = """
import spss.pyspark
from pyspark.sql.types import *

ctx = spss.pyspark.runtime.getContext()

_schema = StructType([StructField('id', LongType(), nullable=False), \
StructField('age', LongType(), nullable=True), \
StructField('Sex', StringType(), nullable=True), \
StructField('BP', StringType(), nullable=True), \
StructField('Cholesterol', StringType(), nullable=True), \
StructField('K', DoubleType(), nullable=True), \
StructField('Na', DoubleType(), nullable=True), \
StructField('Drug', StringType(), nullable=True)])]

if ctx.isComputeDataModelOnly():
    ctx.setSparkOutputSchema(_schema)
else:
    df = ctx.getSparkInputData()
    if df is None:
        drugList=[(1,23,'F','HIGH','HIGH',0.792535,0.031258,'drugY'), \
(2,47,'M','LOW','HIGH',0.739309,0.056468,'drugC'), \
(3,47,'M','LOW','HIGH',0.697269,0.068944,'drugC'), \
(4,28,'F','NORMAL','HIGH',0.563682,0.072289,'drugX'), \
(5,61,'F','LOW','HIGH',0.559294,0.030998,'drugY'), \
(6,22,'F','NORMAL','HIGH',0.676901,0.078647,'drugX'), \
(7,49,'F','NORMAL','HIGH',0.789637,0.048518,'drugY'), \
(8,41,'M','LOW','HIGH',0.766635,0.069461,'drugC'), \
(9,60,'M','NORMAL','HIGH',0.777205,0.05123,'drugY'), \
(10,43,'M','LOW','NORMAL',0.526102,0.027164,'drugY')]
        sqlcxt = ctx.getSparkSQLContext()
        rdd = ctx.getSparkContext().parallelize(drugList)
        print 'pyspark read data count = '+str(rdd.count())
        df = sqlcxt.createDataFrame(rdd, _schema)

    ctx.setSparkOutputData(df)
"""

node.setPropertyValue("python_syntax", python_script)

```

R example

```

##### Script example for R
node.setPropertyValue("syntax_type", "R")

R_script = """# 'JSON Import' Node v1.0 for IBM SPSS Modeler
# 'RJSONIO' package created by Duncan Temple Lang - http://cran.r-project.org/web/packages/RJSONIO
# 'plyr' package created by Hadley Wickham http://cran.r-project.org/web/packages/plyr
# Node developer: Daniel Savine - IBM Extreme Blue 2014
# Description: This node allows you to import into SPSS a table data from a JSON.
# Install function for packages
packages <- function(x){
  x <- as.character(match.call() [[2]])
  if (!require(x,character.only=TRUE)){
    install.packages(pkgs=x,repos="http://cran.r-project.org")
    require(x,character.only=TRUE)
  }
}
# packages
packages(RJSONIO)
packages(plyr)
### This function is used to generate automatically the dataModel
getMetaData <- function (data) {
  if (dim(data)[1]<=0) {

    print("Warning : modelerData has no line, all fieldStorage fields set to strings")
    getStorage <- function(x){return("string")}

  } else {

    getStorage <- function(x) {
      res <- NULL
      #if x is a factor, typeof will return an integer so we treat the case on the side
      if(is.factor(x)) {
        res <- "string"
      } else {
        res <- switch(typeof(unlist(x)),
                      integer = "integer",
                      double = "real",
                      character = "string",

```

```

        "string")
    }
    return (res)
}

col = vector("list", dim(data)[2])
for (i in 1:dim(data)[2]) {
  col[[i]] <- c(fieldName=names(data[i]),
               fieldLabel="",
               fieldStorage=getStorage(data[i]),
               fieldMeasure="",
               fieldFormat="",
               fieldRole="")
}
mdm<-do.call(cbind,col)
mdm<-data.frame(mdm)
return(mdm)
}
# From JSON to a list
txt <- readLines('C:/test.json')
formatedtxt <- paste(txt, collapse = '')
json.list <- fromJSON(formatedtxt)
# Apply path to json.list
if(strsplit(x='true', split='
',fixed=TRUE)[[1]][1]) {
  path.list <- unlist(strsplit(x='id_array', split=','))
  i = 1
  while(i<length(path.list)+1){
    if(is.null(getElement(json.list, path.list[i]))){
      json.list <- json.list[[1]]
    }else{
      json.list <- getElement(json.list, path.list[i])
      i <- i+1
    }
  }
}
# From list to dataframe via unlisted json
i <-1
filled <- data.frame()
while(i < length(json.list)+ 1){
  unlisted.json <- unlist(json.list[[i]])
  to.fill <- data.frame(t(as.data.frame(unlisted.json, row.names = names(unlisted.json))), stringsAsFactors=FALSE)
  filled <- rbind.fill(filled,to.fill)
  i <- 1 + i
}
# Export to SPSS Modeler Data
modelerData <- filled
print(modelerData)
modelerDataModel <- getMetaData(modelerData)
print(modelerDataModel)

"""

node.setPropertyValue("r_syntax", R_script)

```

Table 1. extensionimportnode properties

extensionimportnode properties	Data type	Property description
syntax_type	R Python	Specify which script runs – R or Python (R is the default).
r_syntax	string	The R scripting syntax to run.
python_syntax	string	The Python scripting syntax to run.

fixedfilenode Properties



The Fixed File node imports data from fixed-field text files—that is, files whose fields are not delimited but start at the same position and are of a fixed length. Machine-generated or legacy data are frequently stored in fixed-field format.

Example

```

node = stream.create("fixedfile", "My node")
node.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node.setPropertyValue("record_len", 32)
node.setPropertyValue("skip_header", 1)
node.setPropertyValue("fields", [[{"Age", 1, 3}, {"Sex", 5, 7}, {"BP", 9, 10}, {"Cholesterol", 12, 22}, {"Na", 24, 25}, {"K", 27, 27}, {"Drug", 29, 32}]])

```

```

node.setPropertyValue("decimal_symbol", "Period")
node.setPropertyValue("lines_to_scan", 30)

```

Table 1. fixedfilenode properties

fixedfilenode properties	Data type	Property description
record_len	number	Specifies the number of characters in each record.
line_oriented	flag	Skips the new-line character at the end of each record.
decimal_symbol	Default Comma Period	The type of decimal separator used in your data source.
skip_header	number	Specifies the number of lines to ignore at the beginning of the first record. Useful for ignoring column headers.
auto_recognize_datetime	flag	Specifies whether dates or times are automatically identified in the source data.
lines_to_scan	number	
fields	list	Structured property.
full_filename	string	Full name of file to read, including directory.
strip_spaces	None Left Right Both	Discards leading and trailing spaces in strings on import.
invalid_char_mode	Discard Replace	Removes invalid characters (null, 0, or any character non-existent in current encoding) from the data input or replaces invalid characters with the specified one-character symbol.
invalid_char_replacement	string	
use_custom_values	flag	
custom_storage	Unknown String Integer Real Time Date Timestamp	
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY"	This property is applicable only if a custom storage has been specified.
	"DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" " (H) H: (M) M: (S) S" "(H) H: (M) M" "(M) M: (S) S" "HH.MM.SS" "HH.MM" "MM.SS" "(H) H. (M) M. (S) S" "(H) H. (M) M" "(M) M. (S) S"	This property is applicable only if a custom storage has been specified.
custom_decimal_symbol	field	Applicable only if a custom storage has been specified.
encoding	StreamDefault SystemDefault "UTF-8"	Specifies the text-encoding method.

gsdata_import Node Properties



Use the Geospatial source node to bring map or spatial data into your data mining session.

Table 1. gsdata_import node properties

gsdata_import node properties	Data type	Property description
full_filename	string	Enter the file path to the .shp file you want to load.
map_service_URL	string	Enter the map service URL to connect to.
map_name	string	Only if map_service_URL is used; this contains the top level folder structure of the map service.

Related information

- [Node properties overview](#)
- [Source node common properties](#)
- [databasenode properties](#)
- [datacollectionimportnode Properties](#)
- [excelimportnode Properties](#)
- [evimportnode Properties](#)
- [fixedfilenode Properties](#)
- [sasimportnode Properties](#)
- [simgennode properties](#)
- [userinputnode properties](#)
- [variablefilenode Properties](#)
- [xmlimportnode Properties](#)
- [statisticsimportnode Properties](#)

jsonimportnode Properties



The JSON source node imports data from a JSON file. See [JSON Source node](#) for more information.

Table 1. jsonimportnode properties

jsonimportnode properties	Data type	Property description
<code>full_filename</code>	<code>string</code>	The complete filename, including path.
<code>string_format</code>	<code>records values</code>	Specify the format of the JSON string. Default is <code>records</code> .
<code>auto_label</code>		Added in version 18.2.1.1.

sasimportnode Properties



The SAS Import node imports SAS data into IBM® SPSS® Modeler.

Example

```
node = stream.create("sasimport", "My node")
node.setPropertyValue("format", "Windows")
node.setPropertyValue("full_filename", "C:/data/retail.sas7bdat")
node.setPropertyValue("member_name", "Test")
node.setPropertyValue("read_formats", False)
node.setPropertyValue("full_format_filename", "Test")
node.setPropertyValue("import_names", True)
```

Table 1. sasimportnode properties

sasimportnode properties	Data type	Property description
<code>format</code>	<code>Windows UNIX Transport SAS7 SAS8 SAS9</code>	Format of the file to be imported.
<code>full_filename</code>	<code>string</code>	The complete filename that you enter, including its path.
<code>member_name</code>	<code>string</code>	Specify the member to import from the specified SAS transport file.
<code>read_formats</code>	<code>flag</code>	Reads data formats (such as variable labels) from the specified format file.
<code>full_format_filename</code>	<code>string</code>	
<code>import_names</code>	<code>NamesAndLabels LabelsasNames</code>	Specifies the method for mapping variable names and labels on import.

Related information

- [Node properties overview](#)
- [Source node common properties](#)
- [cognosimport Node Properties](#)
- [databasenode properties](#)
- [datacollectionimportnode Properties](#)
- [excelimportnode Properties](#)

- [evimportnode Properties](#)
- [fixedfilenode Properties](#)
- [simgennode properties](#)
- [userinputnode properties](#)
- [variablefilenode Properties](#)
- [xmlimportnode Properties](#)
- [statisticsimportnode Properties](#)

simgennode properties



The Simulation Generate node provides an easy way to generate simulated data—either from scratch using user specified statistical distributions or automatically using the distributions obtained from running a Simulation Fitting node on existing historical data. This is useful when you want to evaluate the outcome of a predictive model in the presence of uncertainty in the model inputs.

Table 1. simgennode properties

simgennode properties	Data type	Property description
fields	Structured property	See example
correlations	Structured property	See example
keep_min_max_setting	<i>boolean</i>	
refit_correlations	<i>boolean</i>	
max_cases	<i>integer</i>	Minimum value is 1000, maximum value is 2,147,483,647
create_iteration_field	<i>boolean</i>	
iteration_field_name	<i>string</i>	
replicate_results	<i>boolean</i>	
random_seed	<i>integer</i>	
parameter_xml	<i>string</i>	Returns the parameter Xml as a string

fields example

This is a structured slot parameter with the following syntax:

```
simgennode.setPropertyValue("fields", [
    [field1, storage, locked, [distribution1], min, max],
    [field2, storage, locked, [distribution2], min, max],
    [field3, storage, locked, [distribution3], min, max]
])
```

distribution is a declaration of the distribution name followed by a list containing pairs of attribute names and values. Each distribution is defined in the following way:

```
[distributionname, [[par1], [par2], [par3]]]

simgennode = modeler.script.stream().createAt("simgen", u"Sim Gen", 726, 322)
simgennode.setPropertyValue("fields", [{"Age": "integer", "False": ["Uniform", [{"min": "1"}, {"max": "2"}]], "", ""}])
```

For example, to create a node that generates a single field with a Binomial distribution, you might use the following script:

```
simgen_node1 = modeler.script.stream().createAt("simgen", u"Sim Gen", 200, 200)
simgen_node1.setPropertyValue("fields", [{"Education": "Real", "False": ["Binomial", [{"n": 32}, {"prob": 0.7}]], "", ""}])
```

The Binomial distribution takes 2 parameters: **n** and **prob**. Since Binomial does not support minimum and maximum values, these are supplied as an empty string.

Note: You cannot set the **distribution** directly; you use it in conjunction with the **fields** property.

The following examples show all the possible distribution types. Note that the threshold is entered as **thresh** in both **NegativeBinomialFailures** and **NegativeBinomialTrial**.

```
stream = modeler.script.stream()

simgennode = stream.createAt("simgen", u"Sim Gen", 200, 200)

beta_dist = ["Field1", "Real", False, ["Beta", [{"shape1": "1"}, {"shape2": "2"}]], "", ""]
binomial_dist = ["Field2", "Real", False, ["Binomial", [{"n": "1"}, {"prob": "1"}]], "", ""]
categorical_dist = ["Field3", "String", False, ["Categorical", [{"A": "0.3"}, {"B": "0.5"}, {"C": "0.2"}]], "", ""]
dice_dist = ["Field4", "Real", False, ["Dice", [{"1": "0.5"}, {"2": "0.5"}]], "", ""]
exponential_dist = ["Field5", "Real", False, ["Exponential", [{"scale": "1"}]], "", ""]
```

```

fixed_dist = ["Field6", "Real", False, ["Fixed", [[{"value","1"}]], "", ""]
gamma_dist = ["Field7", "Real", False, ["Gamma", [{"scale","1"}, {"shape","1"}]], "", ""]
lognormal_dist = ["Field8", "Real", False, ["Lognormal", [{"a","1"}, {"b","1"}]], "", ""]
negbinomialfailures_dist = ["Field9", "Real", False, ["NegativeBinomialFailures", [{"prob","0.5"}, {"thresh","1"}]], "", ""]
negbinomialtrial_dist = ["Field10", "Real", False, ["NegativeBinomialTrials", [{"prob","0.2"}, {"thresh","1"}]], "", ""]
normal_dist = ["Field11", "Real", False, ["Normal", [{"mean","1"}, {"stddev","2"}]], "", ""]
poisson_dist = ["Field12", "Real", False, ["Poisson", [{"mean","1"}]], "", ""]
range_dist = ["Field13", "Real", False, ["Range", [{"BEGIN","[1,3]"}, {"END","[2,4]"}, {"PROB","[[0.5, 0.5]]"}]], "", ""]
triangular_dist = ["Field14", "Real", False, ["Triangular", [{"min","0"}, {"max","1"}, {"mode","1"}]], "", ""]
uniform_dist = ["Field15", "Real", False, ["Uniform", [{"min","1"}, {"max","2"}]], "", ""]
weibull_dist = ["Field16", "Real", False, ["Weibull", [{"a","0"}, {"b","1"}, {"c","1"}]], "", ""]

simgennode.setPropertyValue("fields", [
beta_dist, \
binomial_dist, \
categorical_dist, \
dice_dist, \
exponential_dist, \
fixed_dist, \
gamma_dist, \
lognormal_dist, \
negbinomialfailures_dist, \
negbinomialtrial_dist, \
normal_dist, \
poisson_dist, \
range_dist, \
triangular_dist, \
uniform_dist, \
weibull_dist
])

```

correlations example

This is a structured slot parameter with the following syntax:

```

simgennode.setPropertyValue("correlations", [
    [field1, field2, correlation],
    [field1, field3, correlation],
    [field2, field3, correlation]
])

```

Correlation can be any number between +1 and -1. You can specify as many or as few correlations as you like. Any unspecified correlations are set to zero. If any fields are unknown, the correlation value should be set on the correlation matrix (or table) and is shown in red text. When there are unknown fields, it is not possible to execute the node.

statisticsimportnode Properties



The IBM® SPSS® Statistics File node reads data from the .sav file format used by IBM SPSS Statistics, as well as cache files saved in IBM SPSS Modeler, which also use the same format.

The properties for this node are described under [statisticsimportnode Properties](#).

tm1odataimport Node Properties



The IBM Cognos TM1 source node imports data from Cognos TM1 databases.

Table 1. tm1odataimport node properties

tm1odataimport node properties	Data type	Property description
credential_type	inputCredential or storedCredential	Used to indicate the credential type.

tm1odataimport rt node properties	Data type	Property description
input_credential	<i>list</i>	When the credential_type is inputCredential ; specify the domain, user name and password.
stored_credential_name	<i>string</i>	When the credential_type is storedCredential ; specify the name of credential on the C&DS server.
selected_view	<i>["field" "field"]</i>	A list property containing the details of the selected TM1 cube and the name of the cube view from where data will be imported into SPSS. For example: <code>TM1_import.setPropertyValue("selected_view", ['plan_BudgetPlan', 'Goal Input'])</code>
is_private_view	<i>flag</i>	Specifies whether the selected_view is a private view. Default value is false .
selected_columns	<i>["field"]</i>	Specify the selected column; only one item can be specified. For example: <code>setPropertyValue("selected_columns", ["Measures"])</code>
selected_rows	<i>["field" "field"]</i>	Specify the selected rows. For example: <code>setPropertyValue("selected_rows", ["Dimension_1_1", "Dimension_2_1", "Dimension_3_1", "Periods"])</code>
connection_type	AdminServer TM1Server	Indicates the connection type. Default is AdminServer .
admin_host	<i>string</i>	The URL for the host name of the REST API. Required if the connection_type is AdminServer .
server_name	<i>string</i>	The name of the TM1 server selected from the admin_host . Required if the connection_type is AdminServer .
server_url	<i>string</i>	The URL for the TM1 Server REST API. Required if the connection_type is TM1Server .

tm1import Node Properties (deprecated)



The IBM Cognos TM1 source node imports data from Cognos TM1 databases.

Note: This node was deprecated in Modeler 18.0. The replacement node script name is *tm1odataimport*.

Table 1. tm1import node properties

tm1import node properties	Data type	Property description
pm_host	<i>string</i>	Note: Only for version 16.0 and 17.0 The host name. For example: <code>TM1_import.setPropertyValue("pm_host", 'http://9.191.86.82:9510/pmhub/pm')</code>
tm1_connection	<i>["field","field", ... , "field"]</i>	Note: Only for version 16.0 and 17.0 A list property containing the connection details for the TM1 server. The format is: ["TM1_Server_Name", "tm1_username", "tm1_password"] For example: <code>TM1_import.setPropertyValue("tm1_connection", ['Planning Sample', 'admin', 'apple'])</code>
selected_view	<i>["field" "field"]</i>	A list property containing the details of the selected TM1 cube and the name of the cube view from where data will be imported into SPSS. For example: <code>TM1_import.setPropertyValue("selected_view", ['plan_BudgetPlan', 'Goal Input'])</code>
selected_columns	<i>["field"]</i>	Specify the selected column; only one item can be specified. For example: <code>setPropertyValue("selected_columns", ["Measures"])</code>
selected_rows	<i>["field" "field"]</i>	Specify the selected rows. For example: <code>setPropertyValue("selected_rows", ["Dimension_1_1", "Dimension_2_1", "Dimension_3_1", "Periods"])</code>

Related information

- [Node properties overview](#)
- [Source node common properties](#)
- [databasenode Properties](#)
- [datacollectionimportnode Properties](#)
- [excelimportnode Properties](#)
- [evimportnode Properties](#)
- [fixedfilenode Properties](#)
- [sasimportnode Properties](#)
- [simgennode Properties](#)
- [userinputnode Properties](#)
- [variablefilenode Properties](#)
- [xmlimportnode Properties](#)
- [statisticsimportnode Properties](#)

twcimport node properties



The TWC source node imports weather data from The Weather Company, an IBM Business. You can use it to obtain historical or forecast weather data for a location. This can help you develop weather-driven business solutions for better decision-making using the most accurate and precise weather data available.

Table 1. twcimport node properties

twcimport node properties	Data type	Property description
TWCDataImport.latitude	Real	Specifies a latitude value in the format [-90.090.0]
TWCDataImport.longitude	Real	Specifies a longitude value in the format [-180.0180.0].
TWCDataImport.licenseKey	string	Specifies the license key obtained from The Weather Company.
TWCDataImport.measurementUnit	English Metric Hybrid	Specifies the measurement unit. Possible values are English, Metric, or Hybrid. Metric is the default.
TWCDataImport.dataType	Historical Forecast	Specifies the type of weather data to input. Possible values are Historical or Forecast. Historical is the default.
TWCDataImport.startDate	Integer	If Historical is specified for <code>TWCDataImport.dataType</code> , specify a start date in the format yyyyMMdd.
TWCDataImport.endDate	Integer	If Historical is specified for <code>TWCDataImport.dataType</code> , specify an end date in the format yyyyMMdd.
TWCDataImport.forecastHour	6 12 24 48	If Forecast is specified for <code>TWCDataImport.dataType</code> , specify 6, 12, 24, or 48 for the hour.

userinputnode properties



The User Input node provides an easy way to create synthetic data—either from scratch or by altering existing data. This is useful, for example, when you want to create a test dataset for modeling.

Example

```
node = stream.create("userinput", "My node")
node.setPropertyValue("names", ["test1", "test2"])
node.setKeyedPropertyValue("data", "test1", "2, 4, 8")
node.setKeyedPropertyValue("custom_storage", "test1", "Integer")
node.setPropertyValue("data_mode", "Ordered")
```

Table 1. userinputnode properties

userinputnode properties	Data type	Property description
data		
names		Structured slot that sets or returns a list of field names generated by the node.
custom_storage	Unknown String Integer Real Time Date Timestamp	Keyed slot that sets or returns the storage for a field.
data_mode	Combined Ordered	If <code>Combined</code> is specified, records are generated for each combination of set values and min/max values. The number of records generated is equal to the product of the number of values in each field. If <code>Ordered</code> is specified, one value is taken from each column for each record in order to generate a row of data. The number of records generated is equal to the largest number values associated with a field. Any fields with fewer data values will be padded with null values.
values		Note: This property has been deprecated in favor of <code>userinputnode.data</code> and should no longer be used.

variablefilenode Properties



The Variable File node reads data from free-field text files—that is, files whose records contain a constant number of fields but a varied number of characters. This node is also useful for files with fixed-length header text and certain types of annotations.

Example

```

node = stream.create("variablefile", "My node")
node.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node.setPropertyValue("read_field_names", True)
node.setPropertyValue("delimit_other", True)
node.setPropertyValue("other", ",")
node.setPropertyValue("quotes_1", "Discard")
node.setPropertyValue("decimal_symbol", "Comma")
node.setPropertyValue("invalid_char_mode", "Replace")
node.setPropertyValue("invalid_char_replacement", "|")
node.setKeyedPropertyValue("use_custom_values", "Age", True)
node.setKeyedPropertyValue("direction", "Age", "Input")
node.setKeyedPropertyValue("type", "Age", "Range")
node.setKeyedPropertyValue("values", "Age", [1, 100])

```

Table 1. variablefilenode properties

variablefile node properties	Data type	Property description
skip_header	number	Specifies the number of characters to ignore at the beginning of the first record.
num_fields_auto	flag	Determines the number of fields in each record automatically. Records must be terminated with a new-line character.
num_fields	number	Manually specifies the number of fields in each record.
delimit_space	flag	Specifies the character used to delimit field boundaries in the file.
delimit_tab	flag	
delimit_new_line	flag	
delimit_non_printing	flag	
delimit_comma	flag	In cases where the comma is both the field delimiter and the decimal separator for streams, set delimit_other to <i>true</i> , and specify a comma as the delimiter by using the other property.
delimit_other	flag	Allows you to specify a custom delimiter using the other property.
other	string	Specifies the delimiter used when delimit_other is <i>true</i> .
decimal_symbol	Default Comma Period	Specifies the decimal separator used in the data source.
multi_blank	flag	Treats multiple adjacent blank delimiter characters as a single delimiter.
read_field_names	flag	Treats the first row in the data file as labels for the column.
strip_spaces	None Left Right Both	Discards leading and trailing spaces in strings on import.
invalid_char_mode	Discard Replace	Removes invalid characters (null, 0, or any character non-existent in current encoding) from the data input or replaces invalid characters with the specified one-character symbol.
invalid_char_replacement	string	
break_case_by_newline	flag	Specifies that the line delimiter is the newline character.
lines_to_scan	number	Specifies how many lines to scan for specified data types.
auto_recognize_datetime	flag	Specifies whether dates or times are automatically identified in the source data.
quotes_1	Discard PairAndDiscard IncludeAsText	Specifies how single quotation marks are treated upon import.
quotes_2	Discard PairAndDiscard IncludeAsText	Specifies how double quotation marks are treated upon import.

variablefile node properties	Data type	Property description
full_filename	string	Full name of file to be read, including directory.
use_custom_values	flag	
custom_storage	Unknown String Integer Real Time Date Timestamp	
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY"	Applicable only if a custom storage has been specified.
	"DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H) H: (M) M: (S) S" "(H) H: (M) M" "(M) M: (S) S" "HH.MM.SS" "HH.MM" "MM.SS" "(H) H. (M) M. (S) S" "(H) H. (M) M" "(M) M. (S) S"	Applicable only if a custom storage has been specified.
custom_decimal_symbol	field	Applicable only if a custom storage has been specified.
encoding	StreamDefault SystemDefault "UTF-8"	Specifies the text-encoding method.

xmlimportnode Properties



The XML source node imports data in XML format into the stream. You can import a single file, or all files in a directory. You can optionally specify a schema file from which to read the XML structure.

Example

```
node = stream.create("xmlimport", "My node")
node.setPropertyValue("full_filename", "c:/import/ebooks.xml")
node.setPropertyValue("records", "/author/name")
```

Table 1. xmlimportnode properties

xmlimportnode properties	Data type	Property description
read	single directory	Reads a single data file (default), or all XML files in a directory.
recurse	flag	Specifies whether to additionally read XML files from all the subdirectories of the specified directory.
full_filename	string	(required) Full path and file name of XML file to import (if read = single).
directory_name	string	(required) Full path and name of directory from which to import XML files (if read = directory).
full_schema_filename	string	Full path and file name of XSD or DTD file from which to read the XML structure. If you omit this parameter, structure is read from the XML source file.
records	string	XPath expression (e.g. /author/name) to define the record boundary. Each time this element is encountered in the source file, a new record is created.
mode	read specify	Read all data (default), or specify which items to read.
fields		List of items (elements and attributes) to import. Each item in the list is an XPath expression.

Related information

- [Node properties overview](#)
- [Source node common properties](#)
- [cognosimport Node Properties](#)
- [databasenode Properties](#)
- [datacollectionimportnode Properties](#)
- [excelimportnode Properties](#)
- [evimportnode Properties](#)
- [fixedfilenode Properties](#)
- [sasimportnode Properties](#)
- [simgennode Properties](#)
- [userinputnode Properties](#)
- [variablefilenode Properties](#)
- [statisticsimporhnode Properties](#)

Record Operations Node Properties

- [appendnode properties](#)
- [aggregatenode properties](#)
- [balancenode properties](#)
- [cplexoptnode properties](#)
- [derive_stbnode properties](#)
- [distinctnode properties](#)
- [extensionprocessnode properties](#)
- [mergenode properties](#)
- [rfmaggregatenode properties](#)
- [samplenode properties](#)
- [selectnode properties](#)
- [sortnode properties](#)
- [spacetimewebs properties](#)
- [streamingtimeseries Properties](#)

appendnode properties



The Append node concatenates sets of records. It is useful for combining datasets with similar structures but different data.

Example

```
node = stream.create("append", "My node")
node.setPropertyValue("match_by", "Name")
node.setPropertyValue("match_case", True)
node.setPropertyValue("include_fields_from", "All")
node.setPropertyValue("create_tag_field", True)
node.setPropertyValue("tag_field_name", "Append_Flag")
```

Table 1. appendnode properties

appendnode properties	Data type	Property description
match_by	Position Name	You can append datasets based on the position of fields in the main data source or the name of fields in the input datasets.
match_case	flag	Enables case sensitivity when matching field names.
include_fields_from	Main All	
create_tag_field	flag	
tag_field_name	string	

aggregatenode properties



The Aggregate node replaces a sequence of input records with summarized, aggregated output records.

Example

```
node = stream.create("aggregate", "My node")
# dbnode is a configured database import node
stream.link(dbnode, node)
node.setPropertyValue("contiguous", True)
node.setPropertyValue("keys", ["Drug"])
node.setKeyedPropertyValue("aggregates", "Age", ["Sum", "Mean"])
node.setPropertyValue("inc_record_count", True)
node.setPropertyValue("count_field", "index")
node.setPropertyValue("extension", "Aggregated_")
node.setPropertyValue("add_as", "Prefix")
```

Table 1. aggregatenode properties

aggregatenode properties	Data type	Property description
--------------------------	-----------	----------------------

aggregatenode properties	Data type	Property description
keys	<i>list</i>	Lists fields that can be used as keys for aggregation. For example, if Sex and Region are your key fields, each unique combination of M and F with regions N and S (four unique combinations) will have an aggregated record.
contiguous	<i>flag</i>	Select this option if you know that all records with the same key values are grouped together in the input (for example, if the input is sorted on the key fields). Doing so can improve performance.
aggregates		Structured property listing the numeric fields whose values will be aggregated, as well as the selected modes of aggregation.
aggregate_exprs		Keyed property which keys the derived field name with the aggregate expression used to compute it. For example: <code>aggregatenode.setKeyedPropertyValue("aggregate_exprs", "Na_MAX", "MAX('Na')")</code>
extension	<i>string</i>	Specify a prefix or suffix for duplicate aggregated fields (sample below).
add_as	<i>Suffix Prefix</i>	
inc_record_count	<i>flag</i>	Creates an extra field that specifies how many input records were aggregated to form each aggregate record.
count_field	<i>string</i>	Specifies the name of the record count field.
allow_approximation	<i>Boolean</i>	Allows approximation of order statistics when aggregation is performed in Analytic Server
bin_count	<i>integer</i>	Specifies the number of bins to use in approximation

balancenode properties

	The Balance node corrects imbalances in a dataset, so it conforms to a specified condition. The balancing directive adjusts the proportion of records where a condition is true by the factor specified.
---	--

Example

```
node = stream.create("balance", "My node")
node.setPropertyValue("training_data_only", True)
node.setPropertyValue("directives", [[1.3, "Age > 60"], [1.5, "Na > 0.5"]])
```

Table 1. balancenode properties

balancenode properties	Data type	Property description
directives		Structured property to balance proportion of field values based on number specified (see example below).
training_data_only	<i>flag</i>	Specifies that only training data should be balanced. If no partition field is present in the stream, then this option is ignored.

This node property uses the format:

`[[number, string] \ [number, string] \ ... [number, string]]`.

Note: If strings (using double quotation marks) are embedded in the expression, they must be preceded by the escape character `" \ "`. The `" \ "` character is also the line continuation character, which you can use to align the arguments for clarity.

cplexoptnode properties

	The CPLEX Optimization node provides the ability to use complex mathematical (CPLEX) based optimization via an Optimization Programming Language (OPL) model file. This functionality was available in the IBM® Analytical Decision Management product, which is no longer supported. But you can also use the CPLEX node in SPSS® Modeler without requiring IBM Analytical Decision Management.
---	--

Table 1. cplexoptnode properties

cplexoptnode properties	Data type	Property description
opl_model_text	<i>string</i>	The OPL (Optimization Programming Language) script program that the CPLEX Optimization node will run and then generate the optimization result.
opl_tuple_set_name	<i>string</i>	The tuple set name in the OPL model that corresponds to the incoming data. This is not required and is normally not set via script. It should only be used for editing field mappings of a selected data source.

cplexoptnode properties	Data type	Property description
<code>data_input_map</code>	<i>List of structured properties</i>	The input field mappings for a data source. This is not required and is normally not set via script. It should only be used for editing field mappings of a selected data source.
<code>md_data_input_map</code>	<i>List of structured properties</i>	<p>The field mappings between each tuple defined in the OPL, with each corresponding field data source (incoming data). Users can edit them each individually per data source. With this script, you can set the property directly to set all mappings at once. This setting is not shown in the user interface.</p> <p>Each entity in the list is structured data:</p> <p>Data Source Tag. The tag of the data source, which can be found in the data source drop-down. For example, for <code>0_Products_Type</code> the tag is <code>0</code>.</p> <p>Data Source Index. The physical sequence (index) of the data source. This is determined by the connection order.</p> <p>Source Node. The source node (annotation) of the data source. This can be found in the data source drop-down. For example, for <code>0_Products_Type</code> the source node is <code>Products</code>.</p> <p>Connected Node. The prior node (annotation) that connects the current CPLEX optimization node. This can be found in the data source drop-down. For example, for <code>0_Products_Type</code> the connected node is <code>Type</code>.</p> <p>Tuple Set Name. The tuple set name of the data source. It must match what's defined in the OPL.</p> <p>Tuple Field Name. The tuple set field name of the data source. It must match what's defined in the OPL tuple set definition.</p> <p>Storage Type. The field storage type. Possible values are <code>int</code>, <code>float</code>, or <code>string</code>.</p>
		<p>Data Field Name. The field name of the data source.</p> <p>Example:</p> <pre>[[0,0,'Product','Type','Products','prod_id_tup','int','prod_id'], [0,0,'Product','Type','Products','prod_name_tup','string', 'prod_name'],[1,1,'Components','Type','Components', 'comp_id_tup','int','comp_id'],[1,1,'Components','Type', 'Components','comp_name_tup','string','comp_name']]</pre>
<code>opl_data_text</code>	<code>string</code>	The definition of some variables or data used for the OPL.
<code>output_value_mode</code>	<code>string</code>	Possible values are <code>raw</code> or <code>dvar</code> . If <code>dvar</code> is specified, on the Output tab the user must specify the object function variable name in OPL for the output. If <code>raw</code> is specified, the objective function will be output directly, regardless of name.
<code>decision_variable_name</code>	<code>string</code>	The objective function variable name in defined in the OPL. This is enabled only when the <code>output_value_mode</code> property is set to <code>dvar</code> .
<code>objective_function_value_fieldname</code>	<code>string</code>	The field name for the objective function value to use in the output. Default is <code>_OBJECTIVE</code> .
<code>output_tuple_set_names</code>	<code>string</code>	<p>The name of the predefined tuples from the incoming data. This acts as the indexes of the decision variable and is expected to be output with the Variable Outputs. The Output Tuple must be consistent with the decision variable definition in the OPL. If there are multiple indexes, the tuple names must be joined by a comma (,).</p> <p>An example for a single tuple is <code>Products</code>, with the corresponding OPL definition being <code>dvar float+ Production[Products];</code></p> <p>An example for multiple tuples is <code>Products,Components</code>, with the corresponding OPL definition being <code>dvar float+ Production[Products] [Components];</code></p>
<code>decision_output_map</code>	<i>List of structured properties</i>	<p>The field mapping between variables defined in the OPL that will be output and the output fields. Each entity in the list is structured data:</p> <p>Variable Name. The variable name in the OPL to output.</p> <p>Storage Type. Possible values are <code>int</code>, <code>float</code>, or <code>string</code>.</p> <p>Output Field Name. The expected field name in the results (output or export).</p> <p>Example:</p> <pre>[['Production','int','res'], ['Remark','string','res_1'],['Cost', 'float','res_2']]</pre>

derive_stbnode properties



The Space-Time-Boxes node derives Space-Time-Boxes from latitude, longitude and timestamp fields. You can also identify frequent Space-Time-Boxes as hangouts.

Example

```
node = modeler.script.stream().createAt("derive_stb", "My node", 96, 96)

# Individual Records mode
node.setPropertyValue("mode", "IndividualRecords")
node.setPropertyValue("latitude_field", "Latitude")
node.setPropertyValue("longitude_field", "Longitude")
node.setPropertyValue("timestamp_field", "OccurredAt")
node.setPropertyValue("densities", ["STB_GH7_1HOUR", "STB_GH7_30MINS"])
node.setPropertyValue("add_extension_as", "Prefix")
node.setPropertyValue("name_extension", "stb_")

# Hangouts mode
node.setPropertyValue("mode", "Hangouts")
node.setPropertyValue("hangout_density", "STB_GH7_30MINS")
node.setPropertyValue("id_field", "Event")
node.setPropertyValue("qualifying_duration", "30MINUTES")
node.setPropertyValue("min_events", 4)
node.setPropertyValue("qualifying_pct", 65)
```

Table 1. Space-Time-Boxes node properties

derive_stbnode properties	Data type	Property description
mode	IndividualRecords Hangouts	
latitude_field	field	
longitude_field	field	
timestamp_field	field	
hangout_density	density	A single density. See densities for valid density values.
densities	[density,density,..,density]	Each density is a string, for example STB_GH8_1DAY . Note: There are limits to which densities are valid. For the geohash, values from GH1 to GH15 can be used. For the temporal part, the following values can be used: EVER 1YEAR 1MONTH 1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2HOURS 1HOUR 30MINS 15MINS 10MINS 5MINS 2MINS 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SEC
id_field	field	

derive_stbnode_properties	Data type	Property description
qualifying_duration	1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2Hours 1HOUR 30MIN 15MIN 10MIN 5MIN 2MIN 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SECS	Must be a string.
min_events	integer	Minimum valid integer value is 2.
qualifying_pct	integer	Must be in the range of 1 and 100.
add_extension_as	Prefix Suffix	
name_extension	string	

distinctnode properties



The Distinct node removes duplicate records, either by passing the first distinct record to the data stream or by discarding the first record and passing any duplicates to the data stream instead.

Example

```
node = stream.create("distinct", "My node")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("fields", ["Age" "Sex"])
node.setPropertyValue("keys_pre_sorted", True)
```

Table 1. distinctnode properties

distinctnode properties	Data type	Property description
mode	Include Discard	You can include the first distinct record in the data stream, or discard the first distinct record and pass any duplicate records to the data stream instead.
grouping_fields	list	Lists fields used to determine whether records are identical. Note: This property is deprecated from IBM® SPSS® Modeler 16 onwards.
composite_value	Structured slot	See example below.
composite_values	Structured slot	See example below.
inc_record_count	flag	Creates an extra field that specifies how many input records were aggregated to form each aggregate record.
count_field	string	Specifies the name of the record count field.
sort_keys	Structured slot.	Note: This property is deprecated from IBM SPSS Modeler 16 onwards.
default_ascending	flag	
low_distinct_key_count	flag	Specifies that you have only a small number of records and/or a small number of unique values of the key field(s).
keys_pre_sorted	flag	Specifies that all records with the same key values are grouped together in the input.
disable_sql_generation	flag	

Example for composite_value property

The composite_value property has the following general form:

```
node.setKeyedPropertyValue("composite_value", FIELD, FİLLOPTION)
```

FİLLOPTION has the form [FillType, Option1, Option2, ...].

Examples:

```
node.setKeyedPropertyValue("composite_value", "Age", ["First"])
node.setKeyedPropertyValue("composite_value", "Age", ["last"])
node.setKeyedPropertyValue("composite_value", "Age", ["Total"])
node.setKeyedPropertyValue("composite_value", "Age", ["Average"])
node.setKeyedPropertyValue("composite_value", "Age", ["Min"])
node.setKeyedPropertyValue("composite_value", "Age", ["Max"])
node.setKeyedPropertyValue("composite_value", "Date", ["Earliest"])
node.setKeyedPropertyValue("composite_value", "Date", ["Latest"])
node.setKeyedPropertyValue("composite_value", "Code", ["FirstAlpha"])
node.setKeyedPropertyValue("composite_value", "Code", ["LastAlpha"])
```

The custom options require more than one argument, these are added as a list, for example:

```
node.setKeyedPropertyValue("composite_value", "Name", ["MostFrequent", "FirstRecord"])
node.setKeyedPropertyValue("composite_value", "Date", ["LeastFrequent", "LastRecord"])
node.setKeyedPropertyValue("composite_value", "Pending", ["IncludesValue", "T", "F"])
node.setKeyedPropertyValue("composite_value", "Marital", ["FirstMatch", "Married", "Divorced", "Separated"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "Space"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "Comma"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "UnderScore"])
```

Example for composite_values property

The composite_values property has the following general form:

```
node.setPropertyValue("composite_values", [
    [FIELD1, [FILLOPTION1]],
    [FIELD2, [FILLOPTION2]],
    .
    .
])
```

Example:

```
node.setPropertyValue("composite_values", [
    ["Age", ["First"]],
    ["Name", ["MostFrequent", "First"]],
    ["Pending", ["IncludesValue", "T"]],
    ["Marital", ["FirstMatch", "Married", "Divorced", "Separated"]],
    ["Code", ["Concatenate", "Comma"]]
])
```

extensionprocessnode properties



With the Extension Transform node, you can take data from a stream and apply transformations to the data using R scripting or Python for Spark scripting.

Python for Spark example

```
#### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_process", "extension_process")
node.setPropertyValue("syntax_type", "Python")

process_script = """
import spss.pyspark.runtime
from pyspark.sql.types import *

ctx = spss.pyspark.runtime.getContext()

if ctx.isComputeDataModelOnly():
    _schema = StructType([StructField("Age", LongType(), nullable=True), \
        StructField("Sex", StringType(), nullable=True), \
        StructField("BP", StringType(), nullable=True), \
        StructField("Na", DoubleType(), nullable=True), \
        StructField("K", DoubleType(), nullable=True), \
        StructField("Drug", StringType(), nullable=True)])
    ctx.setSparkOutputSchema(_schema)
else:
```

```

df = ctxt.getSparkInputData()
print df.dtypes[:]
_newDF = df.select("Age", "Sex", "BP", "Na", "K", "Drug")
print _newDF.dtypes[:]
ctxt.setSparkOutputData(_newDF)
"""

node.setPropertyValue("python_syntax", process_script)

```

R example

```

#### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_syntax", """day<-as.Date(modelerData$dob, format="%Y-%m-%d")
next_day<-day + 1
modelerData<-cbind(modelerData,next_day)
var1<-c(fieldName="Next day",fieldLabel="",fieldStorage="date",fieldMeasure="",fieldFormat="",
fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)""")

```

Table 1. extensionprocessnode properties

extensionprocessnode properties	Data type	Property description
syntax_type	<i>R Python</i>	Specify which script runs – R or Python (R is the default).
r_syntax	<i>string</i>	The R scripting syntax to run.
python_syntax	<i>string</i>	The Python scripting syntax to run.
use_batch_size	<i>flag</i>	Enable use of batch processing.
batch_size	<i>integer</i>	Specify the number of data records to include in each batch.
convert_flags	<i>StringsAndDoubles LogicalValues</i>	Option to convert flag fields.
convert_missing	<i>flag</i>	Option to convert missing values to the R NA value.
convert_datetime	<i>flag</i>	Option to convert variables with date or datetime formats to R date/time formats.
convert_datetime_class	<i>POSIXct POSIXlt</i>	Options to specify to what format variables with date or datetime formats are converted.

mergenode properties



The Merge node takes multiple input records and creates a single output record containing some or all of the input fields. It is useful for merging data from different sources, such as internal customer data and purchased demographic data.

Example

```

node = stream.create("merge", "My node")
# assume customerdata and salesdata are configured database import nodes
stream.link(customerdata, node)
stream.link(salesdata, node)
node.setPropertyValue("method", "Keys")
node.setPropertyValue("key_fields", ["id"])
node.setPropertyValue("common_keys", True)
node.setPropertyValue("join", "PartialOuter")
node.setKeyedPropertyValue("outer_join_tag", "2", True)
node.setKeyedPropertyValue("outer_join_tag", "4", True)
node.setPropertyValue("single_large_input", True)
node.setPropertyValue("single_large_input_tag", "2")
node.setPropertyValue("use_existing_sort_keys", True)
node.setPropertyValue("existing_sort_keys", [{"id": "Ascending"}])

```

Table 1. mergenode properties

mergenode properties	Data type	Property description
method	<i>Order Keys Condition Rankedcondition</i>	Specify whether records are merged in the order they are listed in the data files, if one or more key fields will be used to merge records with the same value in the key fields, if records will be merged if a specified condition is satisfied, or if each row pairing in the primary and all secondary data sets are to be merged; using the ranking expression to sort any multiple matches into order from low to high.
condition	<i>string</i>	If method is set to Condition , specifies the condition for including or discarding records.
key_fields	<i>list</i>	
common_keys	<i>flag</i>	

mergenode properties	Data type	Property description
join	Inner FullOuter PartialOuter Anti	
outer_join_tag.n	flag	In this property, <i>n</i> is the tag name as displayed in the Select Dataset dialog box. Note that multiple tag names may be specified, as any number of datasets could contribute incomplete records.
single_large_input	flag	Specifies whether optimization for having one input relatively large compared to the other inputs will be used.
single_large_input_tag	string	Specifies the tag name as displayed in the Select Large Dataset dialog box. Note that the usage of this property differs slightly from the outer_join_tag property (flag versus string) because only one input dataset can be specified.
use_existing_sort_keys	flag	Specifies whether the inputs are already sorted by one or more key fields.
existing_sort_keys	[['string', 'Ascending'] \['string', 'Descending']]	Specifies the fields that are already sorted and the direction in which they are sorted.
primary_data_set	string	If method is Rankedcondition , select the primary data set in the merge. This can be considered as the left side of an outer join merge.
rename_duplicate_fields	Boolean	If method is Rankedcondition , and this is set to Y , if the resulting merged data set contains multiple fields with the same name from different data sources the respective tags from the data sources are added at the start of the field column headers.
merge_condition	string	
ranking_expression	string	
Num_matches	integer	The number of matches to be returned, based on the merge_condition and ranking_expression . Minimum 1, maximum 100.

rfmaggrenode properties

	The Recency, Frequency, Monetary (RFM) Aggregate node enables you to take customers' historical transactional data, strip away any unused data, and combine all of their remaining transaction data into a single row that lists when they last dealt with you, how many transactions they have made, and the total monetary value of those transactions.
---	---

Example

```
node = stream.create("rfmaggrenode", "My node")
node.setPropertyValue("relative_to", "Fixed")
node.setPropertyValue("reference_date", "2007-10-12")
node.setPropertyValue("id_field", "CardID")
node.setPropertyValue("date_field", "Date")
node.setPropertyValue("value_field", "Amount")
node.setPropertyValue("only_recent_transactions", True)
node.setPropertyValue("transaction_date_after", "2000-10-01")
```

Table 1. rfmaggrenode properties

rfmaggrenode properties	Data type	Property description
relative_to	Fixed Today	Specify the date from which the recency of transactions will be calculated.
reference_date	date	Only available if Fixed is chosen in relative_to .
contiguous	flag	If your data are presorted so that all records with the same ID appear together in the data stream, selecting this option speeds up processing.
id_field	field	Specify the field to be used to identify the customer and their transactions.
date_field	field	Specify the date field to be used to calculate recency against.
value_field	field	Specify the field to be used to calculate the monetary value.
extension	string	Specify a prefix or suffix for duplicate aggregated fields.
add_as	Suffix Prefix	Specify if the extension should be added as a suffix or a prefix.
discard_low_value_records	flag	Enable use of the discard_records_below setting.

rfmaggregate node properties	Data type	Property description
discard_records_below	<i>number</i>	Specify a minimum value below which any transaction details are not used when calculating the RFM totals. The units of value relate to the value field selected.
only_recent_transactions	<i>flag</i>	Enable use of either the specify_transaction_date or transaction_within_last settings.
specify_transaction_date	<i>flag</i>	
transaction_date_after	<i>date</i>	Only available if specify_transaction_date is selected. Specify the transaction date after which records will be included in your analysis.
transaction_within_last	<i>number</i>	Only available if transaction_within_last is selected. Specify the number and type of periods (days, weeks, months, or years) back from the Calculate Recency relative to date after which records will be included in your analysis.
transaction_scale	Days Weeks Months Years	Only available if transaction_within_last is selected. Specify the number and type of periods (days, weeks, months, or years) back from the Calculate Recency relative to date after which records will be included in your analysis.
save_r2	<i>flag</i>	Displays the date of the second most recent transaction for each customer.
save_r3	<i>flag</i>	Only available if save_r2 is selected. Displays the date of the third most recent transaction for each customer.

samplenode properties



The Sample node selects a subset of records. A variety of sample types are supported, including stratified, clustered, and nonrandom (structured) samples. Sampling can be useful to improve performance, and to select groups of related records or transactions for analysis.

Example

```
/* Create two Sample nodes to extract
   different samples from the same data */

node = stream.create("sample", "My node")
node.setPropertyValue("method", "Simple")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("sample_type", "First")
node.setPropertyValue("first_n", 500)

node = stream.create("sample", "My node")
node.setPropertyValue("method", "Complex")
node.setPropertyValue("stratify_by", ["Sex", "Cholesterol"])
node.setPropertyValue("sample_units", "Proportions")
node.setPropertyValue("sample_size_proportions", "Custom")
node.setPropertyValue("sizes_proportions", [{"M": "High", "Default"}, {"M": "Normal", "Default"}, {"F": "High", 0.3}, {"F": "Normal", 0.3}])
```

Table 1. samplenode properties

samplenode properties	Data type	Property description
method	Simple Complex	
mode	Include Discard	Include or discard records that meet the specified condition.
sample_type	First OneInN RandomPct	Specifies the sampling method.
first_n	<i>integer</i>	Records up to the specified cutoff point will be included or discarded.
one_in_n	<i>number</i>	Include or discard every <i>n</i> th record.
rand_pct	<i>number</i>	Specify the percentage of records to include or discard.
use_max_size	<i>flag</i>	Enable use of the maximum_size setting.
maximum_size	<i>integer</i>	Specify the largest sample to be included or discarded from the data stream. This option is redundant and therefore disabled when First and Include are specified.
set_random_seed	<i>flag</i>	Enables use of the random seed setting.
random_seed	<i>integer</i>	Specify the value used as a random seed.
complex_sample_type	Random Systematic	
sample_units	Proportions Counts	
sample_size_proportions	Fixed Custom Variable	

samplenode properties	Data type	Property description
sample_size_counts	Fixed Custom Variable	
fixed_proportions	number	
fixed_counts	integer	
variable_proportions	field	
variable_counts	field	
use_min_stratum_size	flag	
minimum_stratum_size	integer	This option only applies when a Complex sample is taken with Sample units=Proportions .
use_max_stratum_size	flag	
maximum_stratum_size	integer	This option only applies when a Complex sample is taken with Sample units=Proportions .
clusters	field	
stratify_by	[field1 ... fieldN]	
specify_input_weight	flag	
input_weight	field	
new_output_weight	string	
sizes_proportions	[[string string value] [string string value]...]	If sample_units=proportions and sample_size_proportions=Custom , specifies a value for each possible combination of values of stratification fields.
default_proportion	number	
sizes_counts	[[string string value] [string string value]...]	Specifies a value for each possible combination of values of stratification fields. Usage is similar to sizes_proportions but specifying an integer rather than a proportion.
default_count	number	

selectnode properties



The Select node selects or discards a subset of records from the data stream based on a specific condition. For example, you might select the records that pertain to a particular sales region.

Example

```
node = stream.create("select", "My node")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("condition", "Age < 18")
```

Table 1. selectnode properties

selectnode properties	Data type	Property description
mode	Include Discard	Specifies whether to include or discard selected records.
condition	string	Condition for including or discarding records.

sortnode properties



The Sort node sorts records into ascending or descending order based on the values of one or more fields.

Example

```
node = stream.create("sort", "My node")
node.setPropertyValue("keys", [[{"Age": "Ascending"}, {"Sex": "Descending"}]])
node.setPropertyValue("defaultAscending", False)
node.setPropertyValue("useExistingKeys", True)
node.setPropertyValue("existingKeys", [{"Age": "Ascending"}])
```

Table 1. sortnode properties

sortnode properties	Data type	Property description
keys	<i>list</i>	Specifies the fields you want to sort against. If no direction is specified, the default is used.
default_ascending	<i>flag</i>	Specifies the default sort order.
use_existing_keys	<i>flag</i>	Specifies whether sorting is optimized by using the previous sort order for fields that are already sorted.
existing_keys		Specifies the fields that are already sorted and the direction in which they are sorted. Uses the same format as the keys property.

spacetimeboxes properties



Space-Time-Boxes (STB) are an extension of Geohashed spatial locations. More specifically, an STB is an alphanumeric string that represents a regularly shaped region of space and time.

Table 1. spacetimeboxes properties

spacetimeboxes properties	Data type	Property description
mode	<i>IndividualRecords</i> <i>Hangouts</i>	
latitude_field	<i>field</i>	
longitude_field	<i>field</i>	
timestamp_field	<i>field</i>	
densities	<i>[density, density, density...]</i>	<p>Each density is a string. For example: STB_GH8_1DAY Note there are limits to which densities are valid.</p> <p>For the geohash, values from GH1-GH15 can be used.</p> <p>For the temporal part, the following values can be used:</p> <pre> EVER 1YEAR 1MONTH 1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2HOURS 1HOUR 30MINS 15MINS 10MINS 5MINS 2 MINS 1 MIN 30SECS 15SECS 10SECS 5 SECS 2 SECS 1SEC </pre>
field_name_extension	<i>string</i>	
add_extension_as	<i>Prefix</i> <i>Suffix</i>	
hangout_density	<i>density</i>	Single density (see above)
id_field	<i>field</i>	

spacetimeboxes properties	Data type	Property description
qualifying_duration	1DAY 12HOURS 8HOURS 6HOURS 4HOURS 2HOURS 1HOUR 30MIN 15MIN 10MIN 5MIN 2MIN 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SECS	This must be a string.
min_events	integer	Minimum value is 2
qualifying_pct	integer	Must be in range 1-100

streamingtimeseries Properties



The Streaming Time Series node builds and scores time series models in one step.
Note: This Streaming Time Series node replaces the original Streaming TS node that was deprecated in version 18 of SPSS® Modeler.

Table 1. streamingtimeseries properties

streamingtimeseries Properties	Values	Property description
targets	field	The Streaming Time Series node forecasts one or more targets, optionally using one or more input fields as predictors. Frequency and weight fields are not used. See the topic Common modeling node properties for more information.
candidate_inputs	[field1 ... fieldN]	Input or predictor fields used by the model.
use_period	flag	
date_time_field	field	
input_interval	None Unknown Year Quarter Month Week Day Hour Hour_nonperiod Minute Minute_nonperiod Second Second_nonperiod	
period_field	field	
period_start_value	integer	
num_days_per_week	integer	
start_day_of_week	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	
num_hours_per_day	integer	
start_hour_of_day	integer	
timestamp_increments	integer	
cyclic_increments	integer	
cyclic_periods	list	
output_interval	None Year Quarter Month Week Day Hour Minute Second	
is_same_interval	flag	
cross_hour	flag	
aggregate_and_distribute	list	

streamingtimeProperties	Values	Property description
aggregate_default	Mean Sum Mode Min Max	
distribute_default	Mean Sum	
group_default	Mean Sum Mode Min Max	
missing_imput	Linear_interp Series_mean K_mean K_median Linear_trend	
k_span_points	integer	
use_estimation_period	flag	
estimation_period	Observations Times	
date_estimation	list	Only available if you use <code>date_time_field</code>
period_estimation	list	Only available if you use <code>use_period</code>
observations_type	Latest Earliest	
observations_num	integer	
observations_exclude	integer	
method	ExpertModeler Exsmooth Arima	
expert_modeler_method	ExpertModeler Exsmooth Arima	
consider_seasonal	flag	
detect_outliers	flag	
expert_outlier_additive	flag	
expert_outlier_level_shift	flag	
expert_outlier_innovational	flag	
expert_outlier_level_shift	flag	
expert_outlier_transient	flag	
expert_outlier_seasonal_additive	flag	
expert_outlier_local_trend	flag	
expert_outlier_additive_patch	flag	
consider_newesmodels	flag	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative DampedTrendAdditive DampedTrendMultiplicative MultiplicativeTrendAdditive MultiplicativeSeasonal MultiplicativeTrendMultiplicative MultiplicativeTrend	
futureValue_type_method	Compute specify	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arima.p	integer	
arima.d	integer	
arima.q	integer	
arima.sp	integer	
arima.sd	integer	

streamingtime eseries Properties	Values	Property description
arima.sq	integer	
arima_transformation_type	None SquareRoot NaturalLog	
arima_include_constant	flag	
tf_arima.p.fieldname	integer	For transfer functions.
tf_arima.d.fieldname	integer	For transfer functions.
tf_arima.q.fieldname	integer	For transfer functions.
tf_arima.sp.fieldname	integer	For transfer functions.
tf_arima.sd.fieldname	integer	For transfer functions.
tf_arima.sq.fieldname	integer	For transfer functions.
tf_arima.delays.fieldname	integer	For transfer functions.
tf_arima.transformation_type.fieldname	None SquareRoot NaturalLog	For transfer functions.
arima_detect_outliers	flag	
arima_outlier_additive	flag	
arima_outlier_level_shift	flag	
arima_outlier_innovation_all	flag	
arima_outlier_transient	flag	
arima_outlier_seasonal_additive	flag	
arima_outlier_local_trend	flag	
arima_outlier_additive_patch	flag	
conf_limit_pct	real	
events	fields	
forecastperiods	integer	
extend_records_into_future	flag	
conf_limits	flag	
noise_res	flag	

Field Operations Node Properties

- [anonymizenode properties](#)
- [autodataprepnode properties](#)
- [astimeintervalsnode properties](#)
- [binningnode properties](#)
- [derivenode properties](#)
- [ensemblenode properties](#)
- [fillernode properties](#)
- [filternode properties](#)

- [historynode properties](#)
- [partitionnode properties](#)
- [reclassifynode properties](#)
- [reordernode properties](#)
- [reprojectnode properties](#)
- [restructurenode properties](#)
- [rfmanalysisnode properties](#)
- [setoflagnode properties](#)
- [statisticstransformnode properties](#)
- [timeintervalsnode properties \(deprecated\)](#)
- [transposenode properties](#)
- [typenode properties](#)

anonymizenode properties



The Anonymize node transforms the way field names and values are represented downstream, thus disguising the original data. This can be useful if you want to allow other users to build models using sensitive data, such as customer names or other details.

Example

```
stream = modeler.script.stream()
varfilenode = stream.createAt("variablefile", "File", 96, 96)
varfilenode.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")
node = stream.createAt("anonymize", "My node", 192, 96)
# Anonymize node requires the input fields while setting the values
stream.link(varfilenode, node)
node.setKeyedPropertyValue("enable_anonymize", "Age", True)
node.setKeyedPropertyValue("transformation", "Age", "Random")
node.setKeyedPropertyValue("set_random_seed", "Age", True)
node.setKeyedPropertyValue("random_seed", "Age", 123)
node.setKeyedPropertyValue("enable_anonymize", "Drug", True)
node.setKeyedPropertyValue("use_prefix", "Drug", True)
node.setKeyedPropertyValue("prefix", "Drug", "myprefix")
```

Table 1. anonymizenode properties

anonymizene properties	Data type	Property description
<code>enable_anonymize</code>	<code>flag</code>	When set to <code>True</code> , activates anonymization of field values (equivalent to selecting Yes for that field in the Anonymize Values column).
<code>use_prefix</code>	<code>flag</code>	When set to <code>True</code> , a custom prefix will be used if one has been specified. Applies to fields that will be anonymized by the Hash method and is equivalent to choosing the Custom radio button in the Replace Values dialog box for that field.
<code>prefix</code>	<code>string</code>	Equivalent to typing a prefix into the text box in the Replace Values dialog box. The default prefix is the default value if nothing else has been specified.
<code>transformation</code>	<code>Random Fixed</code>	Determines whether the transformation parameters for a field anonymized by the Transform method will be random or fixed.
<code>set_random_seed</code>	<code>flag</code>	When <code>set_random_seed</code> is set to <code>True</code> , the specified seed value will be used (if <code>transformation</code> is also set to <code>Random</code>).
<code>random_seed</code>	<code>integer</code>	When <code>set_random_seed</code> is set to <code>True</code> , this is the seed for the random number.
<code>scale</code>	<code>number</code>	When <code>transformation</code> is set to <code>Fixed</code> , this value is used for "scale by." The maximum scale value is normally 10 but may be reduced to avoid overflow.
<code>translate</code>	<code>number</code>	When <code>transformation</code> is set to <code>Fixed</code> , this value is used for "translate." The maximum translate value is normally 1000 but may be reduced to avoid overflow.

autodataprepnode properties



The Automated Data Preparation (ADP) node can analyze your data and identify fixes, screen out fields that are problematic or not likely to be useful, derive new attributes when appropriate, and improve performance through intelligent screening and sampling techniques. You can use the node in fully automated fashion, allowing the node to choose and apply fixes, or you can preview the changes before they are made and accept, reject, or amend them as desired.

Example

```
node = stream.create("autodataprep", "My node")
node.setPropertyValue("objective", "Balanced")
```

```

node.setPropertyValue("excluded_fields", "Filter")
node.setPropertyValue("prepare_dates_and_times", True)
node.setPropertyValue("compute_time_until_date", True)
node.setPropertyValue("reference_date", "Today")
node.setPropertyValue("units_for_date_durations", "Automatic")

```

Table 1. autodataprepnode properties

autodataprepnode properties	Data type	Property description
objective	Balanced Speed Accuracy Custom	
custom_fields	flag	If true, allows you to specify target, input, and other fields for the current node. If false, the current settings from an upstream Type node are used.
target	field	Specifies a single target field.
inputs	[field1 ... fieldN]	Input or predictor fields used by the model.
use_frequency	flag	
frequency_field	field	
use_weight	flag	
weight_field	field	
excluded_fields	Filter None	
if_fields_do_not_match	StopExecution ClearAnalysis	
prepare_dates_and_times	flag	Control access to all the date and time fields
compute_time_until_date	flag	
reference_date	Today Fixed	
fixed_date	date	
units_for_date_durations	Automatic Fixed	
fixed_date_units	Years Months Days	
compute_time_until_time	flag	
reference_time	CurrentTime Fixed	
fixed_time	time	
units_for_time_durations	Automatic Fixed	
fixed_date_units	Hours Minutes Seconds	
extract_year_from_date	flag	
extract_month_from_date	flag	
extract_day_from_date	flag	
extract_hour_from_time	flag	
extract_minute_from_time	flag	
extract_second_from_time	flag	
exclude_low_quality_inputs	flag	
exclude_too_many_missing	flag	
maximum_percentage_missing	number	
exclude_too_many_categories	flag	
maximum_number_categories	number	
exclude_if_large_category	flag	
maximum_percentage_category	number	
prepare_inputs_and_target	flag	
adjust_type_inputs	flag	
adjust_type_target	flag	

autodataprepnode properties	Data type	Property description
reorder_nominal_inputs	flag	
reorder_nominal_target	flag	
replace_outliers_inputs	flag	
replace_outliers_target	flag	
replace_missing_continuous_inputs	flag	
replace_missing_continuous_target	flag	
replace_missing_nominal_inputs	flag	
replace_missing_nominal_target	flag	
replace_missing_ordinal_inputs	flag	
replace_missing_ordinal_target	flag	
maximum_values_for_ordinal	number	
minimum_values_for_continuous	number	
outlier_cutoff_value	number	
outlier_method	Replace Delete	
rescale_continuous_inputs	flag	
rescaling_method	MinMax ZScore	
min_max_minimum	number	
min_max_maximum	number	
z_score_final_mean	number	
z_score_final_sd	number	
rescale_continuous_target	flag	
target_final_mean	number	
target_final_sd	number	
transform_select_input_fields	flag	
maximize_association_with_target	flag	
p_value_for_merging	number	
merge_ordinal_features	flag	
merge_nominal_features	flag	
minimum_cases_in_category	number	
bin_continuous_fields	flag	
p_value_for_binning	number	
perform_feature_selection	flag	
p_value_for_selection	number	
perform_feature_construction	flag	
transformed_target_name_extension	string	
transformed_inputs_name_extension	string	
constructed_features_root_name	string	
years_duration_name_extension	string	
months_duration_name_extension	string	
days_duration_name_extension	string	

autodatapreppnode properties	Data type	Property description
hours_duration_name_extension	string	
minutes_duration_name_extension	string	
seconds_duration_name_extension	string	
year_cyclical_name_extension	string	
month_cyclical_name_extension	string	
day_cyclical_name_extension	string	
hour_cyclical_name_extension	string	
minute_cyclical_name_extension	string	
second_cyclical_name_extension	string	

astimeintervalsnode properties



Use the Time Intervals node to specify intervals and derive a new time field for estimating or forecasting. A full range of time intervals is supported, from seconds to years.

Table 1. astimeintervalsnode properties

astimeintervals node properties	Data type	Property description
time_field	field	Can accept only a single continuous field. That field is used by the node as the aggregation key for converting the interval. If an integer field is used here it is considered to be a time index.
dimensions	[field1 field2 ... fieldn]	These fields are used to create individual time series based on the field values.
fields_to_aggregate	[field1 field2 ... fieldn]	These fields are aggregated as part of changing the period of the time field. Any fields not included in this picker are filtered out of the data leaving the node.

binningnode properties



The Binning node automatically creates new nominal (set) fields based on the values of one or more existing continuous (numeric range) fields. For example, you can transform a continuous income field into a new categorical field containing groups of income as deviations from the mean. Once you have created bins for the new field, you can generate a Derive node based on the cut points.

Example

```
node = stream.create("binning", "My node")
node.setPropertyValue("fields", ["Na", "K"])
node.setPropertyValue("method", "Rank")
node.setPropertyValue("fixed_width_name_extension", "_binned")
node.setPropertyValue("fixed_width_add_as", "Suffix")
node.setPropertyValue("fixed_bin_method", "Count")
node.setPropertyValue("fixed_bin_count", 10)
node.setPropertyValue("fixed_bin_width", 3.5)
node.setPropertyValue("tile10", True)
```

Table 1. binningnode properties

binningnode properties	Data type	Property description
fields	[field1 field2 ... fieldn]	Continuous (numeric range) fields pending transformation. You can bin multiple fields simultaneously.
method	FixedWidth EqualCount Rank SDev Optimal	Method used for determining cut points for new field bins (categories).
rcalculate_bins	Always IfNecessary	Specifies whether the bins are recalculated and the data placed in the relevant bin every time the node is executed, or that data is added only to existing bins and any new bins that have been added.

binningnode properties	Data type	Property description
<code>fixed_width_name_extension</code>	<code>string</code>	The default extension is <code>_BIN</code> .
<code>fixed_width_add_as</code>	<code>Suffix Prefix</code>	Specifies whether the extension is added to the end (suffix) or to the start (prefix). The default extension is <code>income_BIN</code> .
<code>fixed_bin_method</code>	<code>Width Count</code>	
<code>fixed_bin_count</code>	<code>integer</code>	Specifies an integer used to determine the number of fixed-width bins (categories) for the new field(s).
<code>fixed_bin_width</code>	<code>real</code>	Value (integer or real) for calculating width of the bin.
<code>equal_count_name_extension</code>	<code>string</code>	The default extension is <code>_TILE</code> .
<code>equal_count_add_as</code>	<code>Suffix Prefix</code>	Specifies an extension, either suffix or prefix, used for the field name generated by using standard p-tiles. The default extension is <code>_TILE plus N</code> , where <code>N</code> is the tile number.
<code>tile4</code>	<code>flag</code>	Generates four quantile bins, each containing 25% of cases.
<code>tile5</code>	<code>flag</code>	Generates five quintile bins.
<code>tile10</code>	<code>flag</code>	Generates 10 decile bins.
<code>tile20</code>	<code>flag</code>	Generates 20 vingtile bins.
<code>tile100</code>	<code>flag</code>	Generates 100 percentile bins.
<code>use_custom_tile</code>	<code>flag</code>	
<code>custom_tile_name_extension</code>	<code>string</code>	The default extension is <code>_TILEN</code> .
<code>custom_tile_add_as</code>	<code>Suffix Prefix</code>	
<code>custom_tile</code>	<code>integer</code>	
<code>equal_count_method</code>	<code>RecordCount ValueSum</code>	The <code>RecordCount</code> method seeks to assign an equal number of records to each bin, while <code>ValueSum</code> assigns records so that the sum of the values in each bin is equal.
<code>tied_values_method</code>	<code>Next Current Random</code>	Specifies which bin tied value data is to be put in.
<code>rank_order</code>	<code>Ascending Descending</code>	This property includes <code>Ascending</code> (lowest value is marked 1) or <code>Descending</code> (highest value is marked 1).
<code>rank_add_as</code>	<code>Suffix Prefix</code>	This option applies to rank, fractional rank, and percentage rank.
<code>rank</code>	<code>flag</code>	
<code>rank_name_extension</code>	<code>string</code>	The default extension is <code>_RANK</code> .
<code>rank_fractional</code>	<code>flag</code>	Ranks cases where the value of the new field equals rank divided by the sum of the weights of the nonmissing cases. Fractional ranks fall in the range of 0–1.
<code>rank_fractional_name_extension</code>	<code>string</code>	The default extension is <code>_F_RANK</code> .
<code>rank_pct</code>	<code>flag</code>	Each rank is divided by the number of records with valid values and multiplied by 100. Percentage fractional ranks fall in the range of 1–100.
<code>rank_pct_name_extension</code>	<code>string</code>	The default extension is <code>_P_RANK</code> .
<code>sdev_name_extension</code>	<code>string</code>	
<code>sdev_add_as</code>	<code>Suffix Prefix</code>	
<code>sdev_count</code>	<code>One Two Three</code>	
<code>optimal_name_extension</code>	<code>string</code>	The default extension is <code>_OPTIMAL</code> .
<code>optimal_add_as</code>	<code>Suffix Prefix</code>	
<code>optimal_supervisor_field</code>	<code>field</code>	Field chosen as the supervisory field to which the fields selected for binning are related.
<code>optimal_merge_bins</code>	<code>flag</code>	Specifies that any bins with small case counts will be added to a larger, neighboring bin.
<code>optimal_small_bin_threshold</code>	<code>integer</code>	
<code>optimal_pre_bin</code>	<code>flag</code>	Indicates that prebinning of dataset is to take place.
<code>optimal_max_bins</code>	<code>integer</code>	Specifies an upper limit to avoid creating an inordinately large number of bins.
<code>optimal_lower_end_point</code>	<code>Inclusive Exclusive</code>	
<code>optimal_first_bin</code>	<code>Unbounded Bounded</code>	

binningnode properties	Data type	Property description
optimal_last_bin	Unbounded Bounded	

derivenode properties



The Derive node modifies data values or creates new fields from one or more existing fields. It creates fields of type formula, flag, nominal, state, count, and conditional.

Example 1

```
# Create and configure a Flag Derive field node
node = stream.create("derive", "My node")
node.setPropertyValue("new_name", "DrugX_Flag")
node.setPropertyValue("result_type", "Flag")
node.setPropertyValue("flag_true", "1")
node.setPropertyValue("flag_false", "0")
node.setPropertyValue("flag_expr", "'Drug' == \"drugX\"")

# Create and configure a Conditional Derive field node
node = stream.create("derive", "My node")
node.setPropertyValue("result_type", "Conditional")
node.setPropertyValue("cond_if_cond", "@OFFSET(\"Age\", 1) = \"Age\"")
node.setPropertyValue("cond_then_expr", "(@OFFSET(\"Age\", 1) = \"Age\") >< @INDEX")
node.setPropertyValue("cond_else_expr", "\"Age\"")
```

Example 2

This script assumes that there are two numeric columns called **XPos** and **YPos** that represent the X and Y coordinates of a point (for example, where an event took place). The script creates a Derive node that computes a geospatial column from the X and Y coordinates representing that point in a specific coordinate system:

```
stream = modeler.script.stream()
# Other stream configuration code
node = stream.createAt("derive", "Location", 192, 96)
node.setPropertyValue("new_name", "Location")
node.setPropertyValue("formula_expr", "[['XPos', 'YPos']]")
node.setPropertyValue("formula_type", "Geospatial")
# Now we have set the general measurement type, define the
# specifics of the geospatial object
node.setPropertyValue("geo_type", "Point")
node.setPropertyValue("has_coordinate_system", True)
node.setPropertyValue("coordinate_system", "ETRS_1989_EPSG_Arctic_zone_5-47")
```

Table 1. derivenode properties

derivenode properties	Data type	Property description
new_name	<i>string</i>	Name of new field.
mode	Single Multiple	Specifies single or multiple fields.
fields	<i>list</i>	Used in Multiple mode only to select multiple fields.
name_extension	<i>string</i>	Specifies the extension for the new field name(s).
add_as	Suffix Prefix	Adds the extension as a prefix (at the beginning) or as a suffix (at the end) of the field name.
result_type	Formula Flag Set State Count Conditional	The six types of new fields that you can create.
formula_expr	<i>string</i>	Expression for calculating a new field value in a Derive node.
flag_expr	<i>string</i>	
flag_true	<i>string</i>	
flag_false	<i>string</i>	
set_default	<i>string</i>	
set_value_cond	<i>string</i>	Structured to supply the condition associated with a given value.
state_on_val	<i>string</i>	Specifies the value for the new field when the On condition is met.
state_off_val	<i>string</i>	Specifies the value for the new field when the Off condition is met.

derivenode properties	Data type	Property description
state_on_expression	string	
state_off_expression	string	
state_initial	On Off	Assigns each record of the new field an initial value of on or off . This value can change as each condition is met.
count_initial_val	string	
count_inc_condition	string	
count_inc_expression	string	
count_reset_condition	string	
cond_if_cond	string	
cond_then_expr	string	
cond_else_expr	string	
formula_measure_type	Range / MeasureType.RANGE Discrete / MeasureType.DISCRETE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS Collection / MeasureType.COLLECTION Geospatial / MeasureType.GEOSPATIAL	This property can be used to define the measurement associated with the derived field. The setter function can be passed either a string or one of the MeasureType values. The getter will always return on the MeasureType values.
collection_measure	Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS	For collection fields (lists with a depth of 0), this property defines the measurement type associated with the underlying values.
geo_type	PointMultiPoint LineString MultiLineString Polygon MultiPolygon	For geospatial fields, this property defines the type of geospatial object represented by this field. This should be consistent with the list depth of the values
has_coordinate_system	boolean	For geospatial fields, this property defines whether this field has a coordinate system
coordinate_system	string	For geospatial fields, this property defines the coordinate system for this field

ensemblenode properties



The Ensemble node combines two or more model nuggets to obtain more accurate predictions than can be gained from any one model.

Example

```
# Create and configure an Ensemble node
# Use this node with the models in demos\streams\pm_binaryclassifier.str
node = stream.create("ensemble", "My node")
node.setPropertyValue("ensemble_target_field", "response")
node.setPropertyValue("filter_individual_model_output", False)
node.setPropertyValue("flag_ensemble_method", "ConfidenceWeightedVoting")
node.setPropertyValue("flag_voting_tie_selection", "HighestConfidence")
```

Table 1. ensemblenode properties

ensemblenode properties	Data type	Property description
<code>ensemble_target_field</code>	<code>field</code>	Specifies the target field for all models used in the ensemble.
<code>filter_individual_model_output</code>	<code>flag</code>	Specifies whether scoring results from individual models should be suppressed.
<code>flag_ensemble_method</code>	<code>Voting ConfidenceWeightedVoting RawPropensityWeightedVoting AdjustedPropensityWeightedVoting HighestConfidence AverageRawPropensity AverageAdjustedPropensity</code>	Specifies the method used to determine the ensemble score. This setting applies only if the selected target is a flag field.
<code>set_ensemble_method</code>	<code>Voting ConfidenceWeightedVoting HighestConfidence</code>	Specifies the method used to determine the ensemble score. This setting applies only if the selected target is a nominal field.
<code>flag_voting_tie_selection</code>	<code>Random HighestConfidence RawPropensity AdjustedPropensity</code>	If a voting method is selected, specifies how ties are resolved. This setting applies only if the selected target is a flag field.
<code>set_voting_tie_selection</code>	<code>Random HighestConfidence</code>	If a voting method is selected, specifies how ties are resolved. This setting applies only if the selected target is a nominal field.
<code>calculate_standard_error</code>	<code>flag</code>	If the target field is continuous, a standard error calculation is run by default to calculate the difference between the measured or estimated values and the true values; and to show how close those estimates matched.

fillernode properties



The Filler node replaces field values and changes storage. You can choose to replace values based on a CLEM condition, such as `@BLANK(@FIELD)`. Alternatively, you can choose to replace all blanks or null values with a specific value. A Filler node is often used together with a Type node to replace missing values.

Example

```
node = stream.create("filler", "My node")
node.setPropertyValue("fields", ["Age"])
node.setPropertyValue("replace_mode", "Always")
node.setPropertyValue("condition", "(\"Age\> 60) and (\"Sex\<= \"M\")")
node.setPropertyValue("replace_with", "\"old man\"")
```

Table 1. fillernode properties

fillernode properties	Data type	Property description
<code>fields</code>	<code>list</code>	Fields from the dataset whose values will be examined and replaced.
<code>replace_mode</code>	<code>Always Conditional Blank Null BlankAndNull</code>	You can replace all values, blank values, or null values, or replace based on a specified condition.
<code>condition</code>	<code>string</code>	
<code>replace_with</code>	<code>string</code>	

filternode properties



The Filter node filters (discards) fields, renames fields, and maps fields from one source node to another.

Example:

```
node = stream.create("filter", "My node")
node.setPropertyValue("default_include", True)
node.setKeyedPropertyValue("new_name", "Drug", "Chemical")
node.setKeyedPropertyValue("include", "Drug", False)
```

Using the `default_include` property. Note that setting the value of the `default_include` property does not automatically include or exclude all fields; it simply determines the default for the current selection. This is functionally equivalent to clicking the Include fields by default button in the Filter node dialog box. For example, suppose you run the following script:

```
node = modeler.script.stream().create("filter", "Filter")
node.setPropertyValue("default_include", False)
# Include these two fields in the list
```

```

for f in ["Age", "Sex"]:
    node.setKeyedPropertyValue("include", f, True)

```

This will cause the node to pass the fields **Age** and **Sex** and discard all others. After running the previous script, now suppose you add the following lines to the script to name two more fields:

```

node.setPropertyValue("default_include", False)
# Include these two fields in the list
for f in ["BP", "Na"]:
    node.setKeyedPropertyValue("include", f, True)

```

This will add two more fields to the filter so that a total of four fields are passed (**Age**, **Sex**, **BP**, **Na**). In other words, resetting the value of **default_include** to **False** doesn't automatically reset all fields.

Alternatively, if you now change **default_include** to **True**, either using a script or in the Filter node dialog box, this would flip the behavior so the four fields listed above would be discarded rather than included. When in doubt, experimenting with the controls in the Filter node dialog box may be helpful in understanding this interaction.

Table 1. **filternode** properties

filternode properties	Data type	Property description
default_incl ude	<i>flag</i>	Keyed property to specify whether the default behavior is to pass or filter fields: Note that setting this property does not automatically include or exclude all fields; it simply determines whether selected fields are included or excluded by default. See example below for additional comments.
include	<i>flag</i>	Keyed property for field inclusion and removal.
new_name	<i>string</i>	

historynode properties



The History node creates new fields containing data from fields in previous records. History nodes are most often used for sequential data, such as time series data. Before using a History node, you may want to sort the data using a Sort node.

Example

```

node = stream.create("history", "My node")
node.setPropertyValue("fields", ["Drug"])
node.setPropertyValue("offset", 1)
node.setPropertyValue("span", 3)
node.setPropertyValue("unavailable", "Discard")
node.setPropertyValue("fill_with", "undef")

```

Table 1. **historynode** properties

historynode properties	Data type	Property description
fields	<i>list</i>	Fields for which you want a history.
offset	<i>number</i>	Specifies the latest record (prior to the current record) from which you want to extract historical field values.
span	<i>number</i>	Specifies the number of prior records from which you want to extract values.
unavailable	Discard Leave Fill	For handling records that have no history values, usually referring to the first several records (at the top of the dataset) for which there are no previous records to use as a history.
fill_with	String Number	Specifies a value or string to be used for records where no history value is available.

partitionnode properties



The Partition node generates a partition field, which splits the data into separate subsets for the training, testing, and validation stages of model building.

Example

```

node = stream.create("partition", "My node")
node.setPropertyValue("create_validation", True)
node.setPropertyValue("training_size", 33)
node.setPropertyValue("testing_size", 33)
node.setPropertyValue("validation_size", 33)
node.setPropertyValue("set_random_seed", True)

```

```

node.setPropertyValue("random_seed", 123)
node.setPropertyValue("value_mode", "System")

```

Table 1. partitionnode properties

partitionnode properties	Data type	Property description
<code>new_name</code>	<code>string</code>	Name of the partition field generated by the node.
<code>create_validation</code>	<code>flag</code>	Specifies whether a validation partition should be created.
<code>training_size</code>	<code>integer</code>	Percentage of records (0–100) to be allocated to the training partition.
<code>testing_size</code>	<code>integer</code>	Percentage of records (0–100) to be allocated to the testing partition.
<code>validation_size</code>	<code>integer</code>	Percentage of records (0–100) to be allocated to the validation partition. Ignored if a validation partition is not created.
<code>training_label</code>	<code>string</code>	Label for the training partition.
<code>testing_label</code>	<code>string</code>	Label for the testing partition.
<code>validation_label</code>	<code>string</code>	Label for the validation partition. Ignored if a validation partition is not created.
<code>value_mode</code>	<code>System</code> <code>SystemAndLabel</code> <code>Label</code>	Specifies the values used to represent each partition in the data. For example, the training sample can be represented by the system integer <code>1</code> , the label <code>Training</code> , or a combination of the two, <code>1_Training</code> .
<code>set_random_seed</code>	<code>Boolean</code>	Specifies whether a user-specified random seed should be used.
<code>random_seed</code>	<code>integer</code>	A user-specified random seed value. For this value to be used, <code>set_random_seed</code> must be set to <code>True</code> .
<code>enable_sql_generation</code>	<code>Boolean</code>	Specifies whether to use SQL pushback to assign records to partitions.
<code>unique_field</code>		Specifies the input field used to ensure that records are assigned to partitions in a random but repeatable way. For this value to be used, <code>enable_sql_generation</code> must be set to <code>True</code> .

reclassifynode properties



The Reclassify node transforms one set of categorical values to another. Reclassification is useful for collapsing categories or regrouping data for analysis.

Example

```

node = stream.create("reclassify", "My node")
node.setPropertyValue("mode", "Multiple")
node.setPropertyValue("replace_field", True)
node.setPropertyValue("field", "Drug")
node.setPropertyValue("new_name", "Chemical")
node.setPropertyValue("fields", ["Drug", "BP"])
node.setPropertyValue("name_extension", "reclassified")
node.setPropertyValue("add_as", "Prefix")
node.setKeyedPropertyValue("reclassify", "drugA", True)
node.setPropertyValue("use_default", True)
node.setPropertyValue("default", "BrandX")
node.setPropertyValue("pick_list", ["BrandX", "Placebo", "Generic"])

```

Table 1. reclassifynode properties

reclassifynode properties	Data type	Property description
<code>mode</code>	<code>Single</code> <code>Multiple</code>	<code>Single</code> reclassifies the categories for one field. <code>Multiple</code> activates options enabling the transformation of more than one field at a time.
<code>replace_field</code>	<code>flag</code>	
<code>field</code>	<code>string</code>	Used only in Single mode.
<code>new_name</code>	<code>string</code>	Used only in Single mode.
<code>fields</code>	<code>[field1 field2 ... fieldn]</code>	Used only in Multiple mode.
<code>name_extension</code>	<code>string</code>	Used only in Multiple mode.
<code>add_as</code>	<code>Suffix</code> <code>Prefix</code>	Used only in Multiple mode.
<code>reclassify</code>	<code>string</code>	Structured property for field values.
<code>use_default</code>	<code>flag</code>	Use the default value.
<code>default</code>	<code>string</code>	Specify a default value.

reclassifynode properties	Data type	Property description
pick_list	[string string ... string]	Allows a user to import a list of known new values to populate the drop-down list in the table.

reordernode properties

	The Field Reorder node defines the natural order used to display fields downstream. This order affects the display of fields in a variety of places, such as tables, lists, and the Field Chooser. This operation is useful when working with wide datasets to make fields of interest more visible.
---	--

Example

```
node = stream.create("reorder", "My node")
node.setPropertyValue("mode", "Custom")
node.setPropertyValue("sort_by", "Storage")
node.setPropertyValue("ascending", False)
node.setPropertyValue("start_fields", ["Age", "Cholesterol"])
node.setPropertyValue("end_fields", ["Drug"])
```

Table 1. reordernode properties

reordernode properties	Data type	Property description
mode	Custom Auto	You can sort values automatically or specify a custom order.
sort_by	Name Type Storage	
ascending	flag	
start_fields	[field1 field2 ... fieldn]	New fields are inserted after these fields.
end_fields	[field1 field2 ... fieldn]	New fields are inserted before these fields.

reprojectnode properties

	Within SPSS® Modeler, items such as the Expression Builder spatial functions, the Spatio-Temporal Prediction (STP) Node, and the Map Visualization Node use the projected coordinate system. Use the Reproject node to change the coordinate system of any data that you import that uses a geographic coordinate system.
---	---

Table 1. reprojectnode properties

reprojectnode properties	Data type	Property description
reproject_fileds	[field1 field2 ... fieldn]	List all the fields that are to be reprojected.
reproject_type	Streamdefault Specify	Choose how to reproject the fields.
coordinate_system	string	The name of the coordinate system to be applied to the fields. Example: set reprojectnode.coordinate_system = "WGS_1984_World_Mercator"

restructurenode properties

	The Restructure node converts a nominal or flag field into a group of fields that can be populated with the values of yet another field. For example, given a field named <i>payment type</i> , with values of <i>credit</i> , <i>cash</i> , and <i>debit</i> , three new fields would be created (<i>credit</i> , <i>cash</i> , <i>debit</i>), each of which might contain the value of the actual payment made.
---	---

Example

```
node = stream.create("restructure", "My node")
node.setKeyedPropertyValue("fields_from", "Drug", ["drugA", "drugX"])
node.setPropertyValue("include_field_name", True)
node.setPropertyValue("value_mode", "OtherFields")
node.setPropertyValue("value_fields", ["Age", "BP"])
```

Table 1. restructurenode properties

restructurenode properties	Data type	Property description
-----------------------------------	------------------	-----------------------------

restructurenode properties	Data type	Property description
fields_from	[category category category] all	
include_field_name	flag	Indicates whether to use the field name in the restructured field name.
value_mode	OtherFields Flags	Indicates the mode for specifying the values for the restructured fields. With OtherFields, you must specify which fields to use (see below). With Flags, the values are numeric flags.
value_fields	list	Required if value_mode is OtherFields. Specifies which fields to use as value fields.

rfmanalysisnode properties



The Recency, Frequency, Monetary (RFM) Analysis node enables you to determine quantitatively which customers are likely to be the best ones by examining how recently they last purchased from you (recency), how often they purchased (frequency), and how much they spent over all transactions (monetary).

Example

```
node = stream.create("rfmanalysis", "My node")
node.setPropertyValue("recency", "Recency")
node.setPropertyValue("frequency", "Frequency")
node.setPropertyValue("monetary", "Monetary")
node.setPropertyValue("tied_values_method", "Next")
node.setPropertyValue("recalculate_bins", "IfNecessary")
node.setPropertyValue("recency_thresholds", [1, 500, 800, 1500, 2000, 2500])
```

Table 1. rfmanalysisnode properties

rfmanalysisnode properties	Data type	Property description
recency	field	Specify the recency field. This may be a date, timestamp, or simple number.
frequency	field	Specify the frequency field.
monetary	field	Specify the monetary field.
recency_bins	integer	Specify the number of recency bins to be generated.
recency_weight	number	Specify the weighting to be applied to recency data. The default is 100.
frequency_bins	integer	Specify the number of frequency bins to be generated.
frequency_weight	number	Specify the weighting to be applied to frequency data. The default is 10.
monetary_bins	integer	Specify the number of monetary bins to be generated.
monetary_weight	number	Specify the weighting to be applied to monetary data. The default is 1.
tied_values_method	Next Current	Specify which bin tied value data is to be put in.
recalculate_bins	Always IfNecessary	
add_outliers	flag	Available only if recalculate_bins is set to IfNecessary. If set, records that lie below the lower bin will be added to the lower bin, and records above the highest bin will be added to the highest bin.
binned_field	Recency Frequency Monetary	
recency_thresholds	value value	Available only if recalculate_bins is set to Always. Specify the upper and lower thresholds for the recency bins. The upper threshold of one bin is used as the lower threshold of the next—for example, [10 30 60] would define two bins, the first bin with upper and lower thresholds of 10 and 30, with the second bin thresholds of 30 and 60.
frequency_thresholds	value value	Available only if recalculate_bins is set to Always.
monetary_thresholds	value value	Available only if recalculate_bins is set to Always.

settoflagnode properties

The Set to Flag node derives multiple flag fields based on the categorical values defined for one or more nominal fields.



Example

```
node = stream.create("settoflag", "My node")
node.setKeyedPropertyValue("fields_from", "Drug", ["drugA", "drugX"])
node.setPropertyValue("true_value", "1")
node.setPropertyValue("false_value", "0")
node.setPropertyValue("use_extension", True)
node.setPropertyValue("extension", "Drug_Flag")
node.setPropertyValue("add_as", "Suffix")
node.setPropertyValue("aggregate", True)
node.setPropertyValue("keys", ["Cholesterol"])
```

Table 1. settoflagnode properties

settoflagnode properties	Data type	Property description
fields_from	[category category category] all	
true_value	string	Specifies the true value used by the node when setting a flag. The default is T.
false_value	string	Specifies the false value used by the node when setting a flag. The default is F.
use_extension	flag	Use an extension as a suffix or prefix to the new flag field.
extension	string	
add_as	Suffix Prefix	Specifies whether the extension is added as a suffix or prefix.
aggregate	flag	Groups records together based on key fields. All flag fields in a group are enabled if any record is set to true.
keys	list	Key fields.

statisticstransformnode properties



The Statistics Transform node runs a selection of IBM® SPSS® Statistics syntax commands against data sources in IBM SPSS Modeler. This node requires a licensed copy of IBM SPSS Statistics.

The properties for this node are described under [statisticstransformnode.properties](#).

timeintervalsnode properties (deprecated)



Note: This node was deprecated in version 18 of SPSS® Modeler and replaced by the new Time Series node. The Time Intervals node specifies intervals and creates labels (if needed) for modeling time series data. If values are not evenly spaced, the node can pad or aggregate values as needed to generate a uniform interval between records.

Example

```
node = stream.create("timeintervals", "My node")
node.setPropertyValue("interval_type", "SecondsPerDay")
node.setPropertyValue("days_per_week", 4)
node.setPropertyValue("week_begins_on", "Tuesday")
node.setPropertyValue("hours_per_day", 10)
node.setPropertyValue("day_begins_hour", 7)
node.setPropertyValue("day_begins_minute", 5)
node.setPropertyValue("day_begins_second", 17)
node.setPropertyValue("mode", "Label")
node.setPropertyValue("year_start", 2005)
node.setPropertyValue("month_start", "January")
node.setPropertyValue("day_start", 4)
node.setKeyedPropertyValue("pad", "AGE", "MeanOfRecentPoints")
node.setPropertyValue("agg_mode", "Specify")
node.setPropertyValue("agg_set_default", "Last")
```

Table 1. timeintervalsnode properties

timeinterval snode properties	Data type	Property description

timeinterval snode properties	Data type	Property description
interval_type	None Periods CyclicPeriods Years Quarters Months Days DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic	
mode	Label Create	Specifies whether you want to label records consecutively or build the series based on a specified date, timestamp, or time field.
field	field	When building the series from the data, specifies the field that indicates the date or time for each record.
period_start	integer	Specifies the starting interval for periods or cyclic periods
cycle_start	integer	Starting cycle for cyclic periods.
year_start	integer	For interval types where applicable, year in which the first interval falls.
quarter_start	integer	For interval types where applicable, quarter in which the first interval falls.
month_start	January February March April May June July August September October November December	
day_start	integer	
hour_start	integer	
minute_start	integer	
second_start	integer	
periods_per_cycle	integer	For cyclic periods, number within each cycle.
fiscal_year_begins	January February March April May June July August September October November December	For quarterly intervals, specifies the month when the fiscal year begins.
week_begins_on	Sunday Monday Tuesday Wednesday Thursday Friday Saturday Sunday	For periodic intervals (days per week, hours per day, minutes per day, and seconds per day), specifies the day on which the week begins.
day_begins_hour	integer	For periodic intervals (hours per day, minutes per day, seconds per day), specifies the hour when the day begins. Can be used in combination with <code>day_begins_minute</code> and <code>day_begins_second</code> to specify an exact time such as 8:05:01. See usage example below.
day_begins_minute	integer	For periodic intervals (hours per day, minutes per day, seconds per day), specifies the minute when the day begins (for example, the 5 in 8:05).
day_begins_second	integer	For periodic intervals (hours per day, minutes per day, seconds per day), specifies the second when the day begins (for example, the 17 in 8:05:17).
days_per_week	integer	For periodic intervals (days per week, hours per day, minutes per day, and seconds per day), specifies the number of days per week.
hours_per_day	integer	For periodic intervals (hours per day, minutes per day, and seconds per day), specifies the number of hours in the day.
interval_increment	1 2 3 4 5 6 10 15 20 30	For minutes per day and seconds per day, specifies the number of minutes or seconds to increment for each record.
field_name_extension	string	

timeinterval snode properties	Data type	Property description
field_name_e xtension_as _prefix	flag	
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H) H: (M) M: (S) S" "(H) H: (M) M" (M) M: (S) S "HH.MM.SS" "HH.MM" "MM.SS" "(H) H. (M) M. (S) S" "(H) H. (M) M" (M) M. (S) S	
aggregate	Mean Sum Mode Min Max First Last TrueIfAnyTrue	Specifies the aggregation method for a field.
pad	Blank MeanOfRecentPoints True False	Specifies the padding method for a field.
agg_mode	All Specify	Specifies whether to aggregate or pad all fields with default functions as needed or specify the fields and functions to use.
agg_range_de fault	Mean Sum Mode Min Max	Specifies the default function to use when aggregating continuous fields.
agg_set_defa ult	Mode First Last	Specifies the default function to use when aggregating nominal fields.
agg_flag_def ault	TrueIfAnyTrue Mode First Last	
pad_range_de fault	Blank MeanOfRecentPoints	Specifies the default function to use when padding continuous fields.
pad_set_defa ult	Blank MostRecentValue	
pad_flag_def ault	Blank True False	
max_records_ to_create	integer	Specifies the maximum number of records to create when padding the series.
estimation_f rom_beginning	flag	
estimation_t o_end	flag	
estimation_s tart_offset	integer	

timeinterval snode properties	Data type	Property description
estimation_n um_holdouts	integer	
create_futur e_records	flag	
num_future_r ecords	integer	
create_futur e_field	flag	
future_field _name	string	

transposenode properties



The Transpose node swaps the data in rows and columns so that records become fields and fields become records.

Example

```
node = stream.create("transpose", "My node")
node.setPropertyValue("transposed_names", "Read")
node.setPropertyValue("read_from_field", "TimeLabel")
node.setPropertyValue("max_num_fields", "1000")
node.setPropertyValue("id_field_name", "ID")
```

Table 1. transposenode properties

transposenode properties	Data type	Property description
transpose_method	enum	Specifies the transpose method: Normal (<code>normal</code>), CASE to VAR (<code>casetovar</code>), or VAR to CASE (<code>vartocase</code>).
transposed_names	Prefix Read	Property for the Normal transpose method. New field names can be generated automatically based on a specified prefix, or they can be read from an existing field in the data.
prefix	string	Property for the Normal transpose method.
num_new_fields	integer	Property for the Normal transpose method. When using a prefix, specifies the maximum number of new fields to create.
read_from_field	field	Property for the Normal transpose method. Field from which names are read. This must be an instantiated field or an error will occur when the node is executed.
max_num_fields	integer	Property for the Normal transpose method. When reading names from a field, specifies an upper limit to avoid creating an inordinately large number of fields.
transpose_type	Numeric String Custom	Property for the Normal transpose method. By default, only continuous (numeric range) fields are transposed, but you can choose a custom subset of numeric fields or transpose all string fields instead.
transpose_fields	list	Property for the Normal transpose method. Specifies the fields to transpose when the <code>Custom</code> option is used.
id_field_name	field	Property for the Normal transpose method.
transpose_caseto var_idfields	field	Property for the CASE to VAR (<code>casetovar</code>) transpose method. Accepts multiple fields to be used as index fields. <code>field1 ... fieldN</code>
transpose_caseto var_columnfields	field	Property for the CASE to VAR (<code>casetovar</code>) transpose method. Accepts multiple fields to be used as column fields. <code>field1 ... fieldN</code>
transpose_caseto var_valuefields	field	Property for the CASE to VAR (<code>casetovar</code>) transpose method. Accepts multiple fields to be used as value fields. <code>field1 ... fieldN</code>
transpose_vartoc ase_idfields	field	Property for the VAR to CASE (<code>vartocase</code>) transpose method. Accepts multiple fields to be used as ID variable fields. <code>field1 ... fieldN</code>
transpose_vartoc ase_valfields	field	Property for the VAR to CASE (<code>vartocase</code>) transpose method. Accepts multiple fields to be used as value variable fields. <code>field1 ... fieldN</code>

typenode properties



The Type node specifies field metadata and properties. For example, you can specify a measurement level (continuous, nominal, ordinal, or flag) for each field, set options for handling missing values and system nulls, set the role of a field for

modeling purposes, specify field and value labels, and specify values for a field.

Example

```
node = stream.createAt("type", "My node", 50, 50)
node.setKeyedPropertyValue("check", "Cholesterol", "Coerce")
node.setKeyedPropertyValue("direction", "Drug", "Input")
node.setKeyedPropertyValue("type", "K", "Range")
node.setKeyedPropertyValue("values", "Drug", ["drugA", "drugB", "drugC", "drugD", "drugX",
    "drugY", "drugZ"])
node.setKeyedPropertyValue("null_missing", "BP", False)
node.setKeyedPropertyValue("whitespace_missing", "BP", False)
node.setKeyedPropertyValue("description", "BP", "Blood Pressure")
node.setKeyedPropertyValue("value_labels", "BP", [{"HIGH": "High Blood Pressure"}, {"NORMAL": "normal blood pressure"}])
```

Note that in some cases you may need to fully instantiate the Type node in order for other nodes to work correctly, such as the **fields from** property of the Set to Flag node. You can simply connect a Table node and execute it to instantiate the fields:

```
tablenode = stream.createAt("table", "Table node", 150, 50)
stream.link(node, tablenode)
tablenode.run(None)
stream.delete(tablenode)
```

Table 1. typenode properties

typenode properties	Data type	Property description
direction	Input Target Both None Partition Split Frequency RecordID	Keyed property for field roles. Note: The values In and Out are now deprecated. Support for them may be withdrawn in a future release.
type	Range Flag Set Typeless Discrete OrderedSet Default	Measurement level of the field (previously called the "type" of field). Setting type to Default will clear any values parameter setting, and if value_mode has the value Specify , it will be reset to Read . If value_mode is set to Pass or Read , setting type will not affect value_mode . Note: The data types used internally differ from those visible in the type node. The correspondence is as follows: Range -> Continuous Set -> Nominal OrderedSet -> Ordinal Discrete- > Categorical
storage	Unknown String Integer Real Time Date Timestamp	Read-only keyed property for field storage type.
check	None Nullify Coerce Discard Warn Abort	Keyed property for field type and range checking.
values	[value value]	For continuous fields, the first value is the minimum, and the last value is the maximum. For nominal fields, specify all values. For flag fields, the first value represents <i>false</i> , and the last value represents <i>true</i> . Setting this property automatically sets the value_mode property to Specify .
value_mode	Read Pass Read+ Current Specify	Determines how values are set. Note that you cannot set this property to Specify directly; to use specific values, set the values property.
extend_values	flag	Applies when value_mode is set to Read . Set to T to add newly read values to any existing values for the field. Set to F to discard existing values in favor of the newly read values.
enable_missing	flag	When set to T , activates tracking of missing values for the field.
missing_values	[value value ...]	Specifies data values that denote missing data.
range_missing	flag	Specifies whether a missing-value (blank) range is defined for a field.
missing_lower	string	When range_missing is true, specifies the lower bound of the missing-value range.
missing_upper	string	When range_missing is true, specifies the upper bound of the missing-value range.
null_missing	flag	When set to T , nulls (undefined values that are displayed as \$null\$ in the software) are considered missing values.
whitespace_missing	flag	When set to T , values containing only white space (spaces, tabs, and new lines) are considered missing values.
description	string	Specifies the description for a field.
value_labels	[[Value LabelString] / Value LabelString] ...]	Used to specify labels for value pairs.

typenode properties	Data type	Property description
<code>display_places</code>	<code>integer</code>	Sets the number of decimal places for the field when displayed (applies only to fields with <code>REAL</code> storage). A value of <code>-1</code> will use the stream default.
<code>export_places</code>	<code>integer</code>	Sets the number of decimal places for the field when exported (applies only to fields with <code>REAL</code> storage). A value of <code>-1</code> will use the stream default.
<code>decimal_separator</code>	<code>DEFAULT PERIOD COMMA</code>	Sets the decimal separator for the field (applies only to fields with <code>REAL</code> storage).
<code>date_format</code>	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	Sets the date format for the field (applies only to fields with <code>DATE</code> or <code>TIMESTAMP</code> storage).
<code>time_format</code>	"HHMMSS" "HHMM" "MMSS" "HH : MM : SS" "HH : MM" "MM : SS" "(H) H : (M) M : (S) S" "(H) H : (M) M" "(M) M : (S) S" "HH.MM.SS" "HH.MM" "MM.SS" "(H) H . (M) M . (S) S" "(H) H . (M) M" "(M) M . (S) S"	Sets the time format for the field (applies only to fields with <code>TIME</code> or <code>TIMESTAMP</code> storage).
<code>number_format</code>	<code>DEFAULT STANDARD</code> <code>SCIENTIFIC CURRENCY</code>	Sets the number display format for the field.
<code>standard_places</code>	<code>integer</code>	Sets the number of decimal places for the field when displayed in standard format. A value of <code>-1</code> will use the stream default. Note that the existing <code>display_places</code> slot will also change this but is now deprecated.
<code>scientific_places</code>	<code>integer</code>	Sets the number of decimal places for the field when displayed in scientific format. A value of <code>-1</code> will use the stream default.
<code>currency_places</code>	<code>integer</code>	Sets the number of decimal places for the field when displayed in currency format. A value of <code>-1</code> will use the stream default.
<code>grouping_symbol</code>	<code>DEFAULT NONE LOCALE</code> <code>PERIOD COMMA SPACE</code>	Sets the grouping symbol for the field.
<code>column_width</code>	<code>integer</code>	Sets the column width for the field. A value of <code>-1</code> will set column width to <code>Auto</code> .
<code>justify</code>	<code>AUTO CENTER LEFT</code> <code>RIGHT</code>	Sets the column justification for the field.

typenode properties	Data type	Property description
<code>measure_type</code>	<code>Range / MeasureType.RANGE Discrete / MeasureType.DISCRETE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS S Collection / MeasureType.COLLECTION Geospatial / MeasureType.GEOSPATIAL</code>	This keyed property is similar to <code>type</code> in that it can be used to define the measurement associated with the field. What is different is that in Python scripting, the setter function can also be passed one of the <code>MeasureType</code> values while the getter will always return on the <code>MeasureType</code> values.
<code>collection_measure</code>	<code>Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS</code>	For collection fields (lists with a depth of 0), this keyed property defines the measurement type associated with the underlying values.
<code>geo_type</code>	<code>Point MultiPoint LineString MultiLineString Polygon MultiPolygon</code>	For geospatial fields, this keyed property defines the type of geospatial object represented by this field. This should be consistent with the list depth of the values.
<code>has_coordinate_system</code>	<code>boolean</code>	For geospatial fields, this property defines whether this field has a coordinate system
<code>coordinate_system</code>	<code>string</code>	For geospatial fields, this keyed property defines the coordinate system for this field.
<code>custom_storage_type</code>	<code>Unknown / MeasureType.UNKNOWN String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP List / MeasureType.LIST</code>	This keyed property is similar to <code>custom_storage</code> in that it can be used to define the override storage for the field. What is different is that in Python scripting, the setter function can also be passed one of the <code>StorageType</code> values while the getter will always return on the <code>StorageType</code> values.
<code>custom_list_storage_type</code>	<code>String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP</code>	For list fields, this keyed property specifies the storage type of the underlying values.
<code>custom_list_depth</code>	<code>integer</code>	For list fields, this keyed property specifies the depth of the field
<code>max_list_length</code>	<code>integer</code>	Only available for data with a measurement level of either <i>Geospatial</i> or <i>Collection</i> . Set the maximum length of the list by specifying the number of elements the list can contain.
<code>max_string_length</code>	<code>integer</code>	Only available for <i>typeless</i> data and used when you are generating SQL to create a table. Enter the value of the largest string in your data; this generates a column in the table that is big enough to contain the string.

Graph Node Properties

- [Graph node common properties](#)
- [collectionnode Properties](#)
- [distributionnode Properties](#)
- [evaluationnode Properties](#)
- [graphboardnode Properties](#)
- [histogramnode Properties](#)
- [mapvisualization properties](#)
- [multiplotnode Properties](#)
- [plotnode Properties](#)
- [timeplotnode Properties](#)
- [eplotnode Properties](#)
- [tsnenode Properties](#)
- [webnode Properties](#)

Graph node common properties

This section describes the properties available for graph nodes, including common properties and properties that are specific to each node type.

Table 1. Common graph node properties

Common graph node properties	Data type	Property description
<code>title</code>	<code>string</code>	Specifies the title. Example: "This is a title."
<code>caption</code>	<code>string</code>	Specifies the caption. Example: "This is a caption."
<code>output_mode</code>	<code>Screen File</code>	Specifies whether output from the graph node is displayed or written to a file.
<code>output_format</code>	<code>BMP JPEG PNG HTML output (.cou)</code>	Specifies the type of output. The exact type of output allowed for each node varies.
<code>full_filename</code>	<code>string</code>	Specifies the target path and filename for output generated from the graph node.
<code>use_graph_size</code>	<code>flag</code>	Controls whether the graph is sized explicitly, using the width and height properties below. Affects only graphs that are output to screen. Not available for the Distribution node.
<code>graph_width</code>	<code>number</code>	When <code>use_graph_size</code> is <code>True</code> , sets the graph width in pixels.
<code>graph_height</code>	<code>number</code>	When <code>use_graph_size</code> is <code>True</code> , sets the graph height in pixels.

Turning off optional fields

Optional fields, such as an overlay field for plots, can be turned off by setting the property value to " " (empty string), as shown in the following example:

```
plotnode.setPropertyValue("color_field", "")
```

Specifying colors

The colors for titles, captions, backgrounds, and labels can be specified by using the hexadecimal strings starting with the hash (#) symbol. For example, to set the graph background to sky blue, you would use the following statement:

```
mygraphnode.setPropertyValue("graph_background", "#87CEEB")
```

Here, the first two digits, 87, specify the red content; the middle two digits, CE, specify the green content; and the last two digits, EB, specify the blue content. Each digit can take a value in the range 0–9 or A–F. Together, these values can specify a red-green-blue, or RGB, color.

Note: When specifying colors in RGB, you can use the Field Chooser in the user interface to determine the correct color code. Simply hover over the color to activate a ToolTip with the desired information.

collectionnode Properties

The Collection node shows the distribution of values for one numeric field relative to the values of another. (It creates graphs that are similar to histograms.) It is useful for illustrating a variable or field whose values change over time. Using 3-D



graphing, you can also include a symbolic axis displaying distributions by category.

Example

```
node = stream.create("collection", "My node")
# "Plot" tab
node.setPropertyValue("three_D", True)
node.setPropertyValue("collect_field", "Drug")
node.setPropertyValue("over_field", "Age")
node.setPropertyValue("by_field", "BP")
node.setPropertyValue("operation", "Sum")
# "Overlay" section
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "")
# "Options" tab
node.setPropertyValue("range_mode", "Automatic")
node.setPropertyValue("range_min", 1)
node.setPropertyValue("range_max", 100)
node.setPropertyValue("bins", "ByNumber")
node.setPropertyValue("num_bins", 10)
node.setPropertyValue("bin_width", 5)
```

Table 1. collectionnode properties

collectionnode properties	Data type	Property description
over_field	field	
over_label_auto	flag	
over_label	string	
collect_field	field	
collect_label_auto	flag	
collect_label	string	
three_D	flag	
by_field	field	
by_label_auto	flag	
by_label	string	
operation	Sum Mean Min Max SDev	
color_field	string	
panel_field	string	
animation_field	string	
range_mode	Automatic UserDefined	
range_min	number	
range_max	number	
bins	ByNumber ByWidth	
num_bins	number	
bin_width	number	
use_grid	flag	
graph_background	color	Standard graph colors are described at the beginning of this section.
page_background	color	Standard graph colors are described at the beginning of this section.

Related information

- [distributionnode Properties](#)

distributionnode Properties



The Distribution node shows the occurrence of symbolic (categorical) values, such as mortgage type or gender. Typically, you might use the Distribution node to show imbalances in the data, which you could then rectify using a Balance node before creating a model.

Example

```
node = stream.create("distribution", "My node")
# "Plot" tab
node.setPropertyValue("plot", "Flags")
node.setPropertyValue("x_field", "Age")
```

```

node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("normalize", True)
node.setPropertyValue("sort_mode", "ByOccurrence")
node.setPropertyValue("use_proportional_scale", True)

```

Table 1. distributionnode properties

distributionnode properties	Data type	Property description
plot	SelectedFields Flags	
x_field	field	
color_field	field	Overlay field.
normalize	flag	
sort_mode	ByOccurrence Alphabetic	
use_proportional_scale	flag	

Related information

- [collectionnode Properties](#)

evaluationnode Properties



The Evaluation node helps to evaluate and compare predictive models. The evaluation chart shows how well models predict particular outcomes. It sorts records based on the predicted value and confidence of the prediction. It splits the records into groups of equal size (**quantiles**) and then plots the value of the business criterion for each quantile from highest to lowest. Multiple models are shown as separate lines in the plot.

Example

```

node = stream.create("evaluation", "My node")
# "Plot" tab
node.setPropertyValue("chart_type", "Gains")
node.setPropertyValue("cumulative", False)
node.setPropertyValue("field_detection_method", "Name")
node.setPropertyValue("inc_baseline", True)
node.setPropertyValue("n_tile", "Deciles")
node.setPropertyValue("style", "Point")
node.setPropertyValue("point_type", "Dot")
node.setPropertyValue("use_fixed_cost", True)
node.setPropertyValue("cost_value", 5.0)
node.setPropertyValue("cost_field", "Na")
node.setPropertyValue("use_fixed_revenue", True)
node.setPropertyValue("revenue_value", 30.0)
node.setPropertyValue("revenue_field", "Age")
node.setPropertyValue("use_fixed_weight", True)
node.setPropertyValue("weight_value", 2.0)
node.setPropertyValue("weight_field", "K")

```

Table 1. evaluationnode properties

evaluationnode properties	Data type	Property description
chart_type	Gains Response Lift Profit ROI ROC	
inc_baseline	flag	
field_detection_method	Metadata Name	
use_fixed_cost	flag	
cost_value	number	
cost_field	string	
use_fixed_revenue	flag	
revenue_value	number	
revenue_field	string	
use_fixed_weight	flag	
weight_value	number	
weight_field	field	
n_tile	Quartiles Quintiles Deciles Vingtiles Percentiles 1000-tiles	
cumulative	flag	

evaluationnode properties	Data type	Property description
style	Line Point	
point_type	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
export_data	flag	
data_filename	string	
delimiter	string	
new_line	flag	
inc_field_names	flag	
inc_best_line	flag	
inc_business_rule	flag	
business_rule_condition	string	
plot_score_fields	flag	
score_fields	[field1 ... fieldN]	
target_field	field	
use_hit_condition	flag	
hit_condition	string	
use_score_expression	flag	
score_expression	string	
caption_auto	flag	

Related information

- [Node properties overview](#)
- [Graph node common properties](#)
- [graphboardnode Properties](#)
- [histogramnode Properties](#)
- [multiplotnode Properties](#)
- [plotnode Properties](#)
- [timeplotnode Properties](#)
- [webnode Properties](#)

graphboardnode Properties



The Graphboard node offers many different types of graphs in one single node. Using this node, you can choose the data fields you want to explore and then select a graph from those available for the selected data. The node automatically filters out any graph types that would not work with the field choices.

Note: If you set a property that is not valid for the graph type (for example, specifying **y_field** for a histogram), that property is ignored.

Note: In the UI, on the Detailed tab of many different graph types, there is a Summary field; this field is not currently supported by scripting.

Example

```
node = stream.create("graphboard", "My node")
node.setPropertyValue("graph_type", "Line")
node.setPropertyValue("x_field", "K")
node.setPropertyValue("y_field", "Na")
```

Table 1. graphboardnode properties

graphboard properties	Data type	Property description
graph_type	2DDotplot 3DArea 3DBar 3DDensity 3DHistogram 3DPie 3DScatterplot Area ArrowMap Bar BarCounts BarCountsMap BarMap BinnedScatter Boxplot Bubble ChoroplethMeans ChoroplethMedians ChoroplethSums ChoroplethValues	Identifies the graph type.
	ChoroplethCounts CoordinateMap CoordinateChoroplethMeans CoordinateChoroplethMedians CoordinateChoroplethSums CoordinateChoroplethValues CoordinateChoroplethCounts Dotplot Heatmap HexBinScatter Histogram Line LineChartMap LineOverlayMap Parallel Path Pie PieCountMap PieCounts PieMap	
	PointOverlayMap PolygonOverlayMap Ribbon Scatterplot SPLOM Surface	
x_field	field	Specifies a custom label for the x axis. Available only for labels.
y_field	field	Specifies a custom label for the y axis. Available only for labels.
z_field	field	Used in some 3-D graphs.
color_field	field	Used in heat maps.
size_field	field	Used in bubble plots.
categories_field	field	
values_field	field	
rows_field	field	
columns_field	field	
fields	field	
start_longitude_field	field	Used with arrows on a reference map.
end_longitude_field	field	
start_latitude_field	field	
end_latitude_field	field	
data_key_field	field	Used in various maps.
panelrow_field	string	
panelcol_field	string	
animation_field	string	
longitude_field	field	Used with coordinates on maps.
latitude_field	field	
map_color_field	field	

histogramnode Properties

	The Histogram node shows the occurrence of values for numeric fields. It is often used to explore the data before manipulations and model building. Similar to the Distribution node, the Histogram node frequently reveals imbalances in the data.
---	---

Example

```
node = stream.create("histogram", "My node")
# "Plot" tab
node.setPropertyValue("field", "Drug")
```

```

node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "")
# "Options" tab
node.setPropertyValue("range_mode", "Automatic")
node.setPropertyValue("range_min", 1.0)
node.setPropertyValue("range_max", 100.0)
node.setPropertyValue("num_bins", 10)
node.setPropertyValue("bin_width", 10)
node.setPropertyValue("normalize", True)
node.setPropertyValue("separate_bands", False)

```

Table 1. histogramnode properties

histogramnode properties	Data type	Property description
field	field	
color_field	field	
panel_field	field	
animation_field	field	
range_mode	Automatic UserDefined	
range_min	number	
range_max	number	
bins	ByNumber ByWidth	
num_bins	number	
bin_width	number	
normalize	flag	
separate_bands	flag	
x_label_auto	flag	
x_label	string	
y_label_auto	flag	
y_label	string	
use_grid	flag	
graph_background	color	Standard graph colors are described at the beginning of this section.
page_background	color	Standard graph colors are described at the beginning of this section.
normal_curve	flag	Indicates whether the normal distribution curve should be shown on the output.

Related information

- [Node properties overview](#)
- [Graph node common properties](#)
- [evaluationnode Properties](#)
- [graphboardnode Properties](#)
- [multiplotnode Properties](#)
- [plotnode Properties](#)
- [timeplotnode Properties](#)
- [webnode Properties](#)

mapvisualization properties



The Map Visualization node can accept multiple input connections and display geospatial data on a map as a series of layers. Each layer is a single geospatial field; for example, the base layer might be a map of a country, then above that you might have one layer for roads, one layer for rivers, and one layer for towns.

Table 1. mapvisualization properties

mapvisualization properties	Data type	Property description
tag	string	Sets the name of the tag for the input. The default tag is a number based on the order that inputs were connected to the node (the first connection tag is 1, the second connection tag is 2, etc.).

mapvisualizati on properties	Data type	Property description
layer_field	<i>field</i>	Selects which geo-field from the data set is displayed as a layer on the map. The default selection is based on the following sort order: <ul style="list-style-type: none"> • First - Point • Linestring • Polygon • Multipoint • MultiLinestring • Last - MultiPolygon If there are two fields with the same measurement type, the first field alphabetically (by name) will be selected by default.
color_type	<i>boolean</i>	Specifies whether a standard color is applied to all features of the geo-field, or an overlay field which colors the features based on values from another field in the data set. Possible values are standard or overlay . The default is standard .
color	<i>string</i>	If standard is selected for color_type , the drop-down contains the same color palette as the chart category color order on the user options Display tab. Default is chart category color 1.
color_field	<i>field</i>	If overlay is selected for color_type , the drop-down contains all fields from the same data set as the geo-field selected as the layer.
symbol_type	<i>boolean</i>	Specifies whether a standard symbol is applied to all records of the geo-field, or an overlay symbol which changes the symbol icon for the points based on values from another field in the data set. Possible values are standard or overlay . The default is standard .
symbol	<i>string</i>	If standard is selected for symbol_type , the drop-down contains a selection of symbols that can be used to display points on the map.
symbol_field	<i>field</i>	If overlay is selected for symbol_type , the drop-down contains all of the nominal, ordinal, or categorical fields from the same data set as the geo-field selected as the layer.
size_type	<i>boolean</i>	Specifies whether a standard size is applied to all records of the geo-field, or an overlay size which changes the size of symbol icon or the line thickness based on values from another field in the data set. Possible values are standard or overlay . The default is standard .
size	<i>string</i>	If standard is selected for size_type , for point or multipoint , the drop-down contains a selection of sizes for the symbol selected. For linestring or multilinestring , the drop-down contains a selection of line thicknesses.
size_field	<i>field</i>	If overlay is selected for size_type , the drop-down contains all of the fields from the same data set as the geo-field selected as the layer.
transp_type	<i>boolean</i>	Specifies whether a standard transparency is applied to all records of the geo-field, or an overlay transparency which changes the level of transparency for the symbol, line, or polygon based on values from another field in the data set. Possible values are standard or overlay . The default is standard .
transp	<i>integer</i>	If standard is selected for transp_type , the drop-down contains a selection of transparency levels starting at 0% (opaque) and increasing to 100% (transparent) in 10% increments. Sets the transparency of points, lines, or polygons on the map. If overlay is selected for size_type , the drop-down contains all of the fields from the same data set as the geo-field selected as the layer. For points , multipoints , linestrings , and multilinestrings , polygons and multipolygons (that are the bottom layer), the default is 0%. For polygons and multipolygons that are not the bottom layer, the default is 50% (to avoid obscuring layers beneath these polygons).
transp_field	<i>field</i>	If overlay is selected for transp_type , the drop-down contains all of the fields from the same data set as the geo-field selected as the layer.
data_label_fie ld	<i>field</i>	Specifies the field to use as data labels on the map. For example, if the layer this setting is applied to is a polygon layer, then the data label might be the name field – containing the name of each polygon. So selecting the name field here would result in those names being displayed on the map.
use_hex_binnin g	<i>boolean</i>	Enables hex binning and enables all of the aggregation drop-downs. This setting is turned off by default.

mapvisualizati on properties	Data type	Property description
color_aggregat ion and transp_aggrega tion	string	<p>If you select an overlay field for a points layer using hex binning, then all the values for that field must be aggregated for all points within the hexagon. Therefore, you must specify an aggregation function for any overlay fields you want to apply to the map.</p> <p>The available aggregation functions are:</p> <p>Continuous (Real or Integer storage):</p> <ul style="list-style-type: none"> • Sum • Mean • Min • Max • Median • 1st Quartile • 3rd Quartile <p>Continuous (Time, Date, or Timestamp storage):</p> <ul style="list-style-type: none"> • Mean • Min • Max <p>Nominal/Categorical:</p> <ul style="list-style-type: none"> • Mode • Min • Max <p>Flag:</p> <ul style="list-style-type: none"> • True if any true • False if any false
custom_storage	string	Sets the overall storage type of the field. Default is List . If List is specified, the following custom_value_storage and list_depth controls are disabled.
custom_value_s torage	string	Sets the storage types of the elements in the list instead of to the field as a whole. The default is Real .
list_depth	integer	<p>Sets the depth of the list field. Ther required depth depends on the type of geofield, following these criteria:</p> <ul style="list-style-type: none"> • Point - 0 • LineString - 1 • Polygon - 2 • Multipoint - 1 • MultiLineString - 2 • Multipolygon - 3 <p>You must know the type of geospatial field you are converting back to a list and the required depth for that kind of field. If set incorrectly, the field cannot be used.</p> <p>The default value is 0, minimum is 0, and maximum is 10.</p>

multiplotnode Properties



The Multiplot node creates a plot that displays multiple Y fields over a single X field. The Y fields are plotted as colored lines; each is equivalent to a Plot node with Style set to Line and X Mode set to Sort. Multiplots are useful when you want to explore the fluctuation of several variables over time.

Example

```
node = stream.create("multiplot", "My node")
# "Plot" tab
node.setPropertyValue("x_field", "Age")
node.setPropertyValue("y_fields", ["Drug", "BP"])
node.setPropertyValue("panel_field", "Sex")
# "Overlay" section
node.setPropertyValue("animation_field", "")
node.setPropertyValue("tooltip", "test")
node.setPropertyValue("normalize", True)
```

```

node.setPropertyValue("use_overlay_expr", False)
node.setPropertyValue("overlay_expression", "test")
node.setPropertyValue("records_limit", 500)
node.setPropertyValue("if_over_limit", "PlotSample")

```

Table 1. multiplotnode properties

multiplotnode properties	Data type	Property description
x_field	field	
y_fields	list	
panel_field	field	
animation_field	field	
normalize	flag	
use_overlay_expr	flag	
overlay_expression	string	
records_limit	number	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	flag	
x_label	string	
y_label_auto	flag	
y_label	string	
use_grid	flag	
graph_background	color	Standard graph colors are described at the beginning of this section.
page_background	color	Standard graph colors are described at the beginning of this section.

Related information

- [Node properties overview](#)
- [Graph node common properties](#)
- [evaluationnode Properties](#)
- [graphboardnode Properties](#)
- [histogramnode Properties](#)
- [plotnode Properties](#)
- [timeplotnode Properties](#)
- [webnode Properties](#)

plotnode Properties



The Plot node shows the relationship between numeric fields. You can create a plot by using points (a scatterplot) or lines.

Example

```

node = stream.create("plot", "My node")
# "Plot" tab
node.setPropertyValue("three_D", True)
node.setPropertyValue("x_field", "BP")
node.setPropertyValue("y_field", "Cholesterol")
node.setPropertyValue("z_field", "Drug")
# "Overlay" section
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("size_field", "Age")
node.setPropertyValue("shape_field", "")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "BP")
node.setPropertyValue("transp_field", "")
node.setPropertyValue("style", "Point")
# "Output" tab
node.setPropertyValue("output_mode", "File")
node.setPropertyValue("output_format", "JPEG")
node.setPropertyValue("full_filename", "C:/temp/graph_output/plot_output.jpeg")

```

Table 1. plotnode properties

plotnode properties	Data type	Property description
x_field	field	Specifies a custom label for the x axis. Available only for labels.
y_field	field	Specifies a custom label for the y axis. Available only for labels.

plotnode properties	Data type	Property description
<code>three_D</code>	<code>flag</code>	Specifies a custom label for the <i>y</i> axis. Available only for labels in 3-D graphs.
<code>z_field</code>	<code>field</code>	
<code>color_field</code>	<code>field</code>	Overlay field.
<code>size_field</code>	<code>field</code>	
<code>shape_field</code>	<code>field</code>	
<code>panel_field</code>	<code>field</code>	Specifies a nominal or flag field for use in making a separate chart for each category. Charts are paneled together in one output window.
<code>animation_field</code>	<code>field</code>	Specifies a nominal or flag field for illustrating data value categories by creating a series of charts displayed in sequence using animation.
<code>transp_field</code>	<code>field</code>	Specifies a field for illustrating data value categories by using a different level of transparency for each category. Not available for line plots.
<code>overlay_type</code>	<code>None Smoother Function</code>	Specifies whether an overlay function or LOESS smoother is displayed.
<code>overlay_expression</code>	<code>string</code>	Specifies the expression used when <code>overlay_type</code> is set to <code>Function</code> .
<code>style</code>	<code>Point Line</code>	
<code>point_type</code>	<code>Rectangle</code> <code>Dot</code> <code>Triangle</code> <code>Hexagon</code> <code>Plus</code> <code>Pentagon</code> <code>Star</code> <code>BowTie</code> <code>HorizontalDash</code> <code>VerticalDash</code> <code>IronCross</code> <code>Factory</code> <code>House</code> <code>Cathedral</code> <code>OnionDome</code> <code>ConcaveTriangle</code> <code>OblateGlobe</code> <code>CatEye</code> <code>FourSidedPillow</code> <code>RoundRectangle</code> <code>Fan</code>	
<code>x_mode</code>	<code>Sort Overlay As Read</code>	
<code>x_range_mode</code>	<code>Automatic</code> <code>UserDefined</code>	
<code>x_range_min</code>	<code>number</code>	
<code>x_range_max</code>	<code>number</code>	
<code>y_range_mode</code>	<code>Automatic</code> <code>UserDefined</code>	
<code>y_range_min</code>	<code>number</code>	
<code>y_range_max</code>	<code>number</code>	
<code>z_range_mode</code>	<code>Automatic</code> <code>UserDefined</code>	
<code>z_range_min</code>	<code>number</code>	
<code>z_range_max</code>	<code>number</code>	
<code>jitter</code>	<code>flag</code>	
<code>records_limit</code>	<code>number</code>	
<code>if_over_limit</code>	<code>PlotBins</code> <code>PlotSample PlotAll</code>	
<code>x_label_auto</code>	<code>flag</code>	
<code>x_label</code>	<code>string</code>	
<code>y_label_auto</code>	<code>flag</code>	
<code>y_label</code>	<code>string</code>	
<code>z_label_auto</code>	<code>flag</code>	
<code>z_label</code>	<code>string</code>	
<code>use_grid</code>	<code>flag</code>	
<code>graph_background</code>	<code>color</code>	Standard graph colors are described at the beginning of this section.
<code>page_background</code>	<code>color</code>	Standard graph colors are described at the beginning of this section.

plotnode properties	Data type	Property description
use_overlay_expr	flag	Deprecated in favor of <code>overlay_type</code> .

timeplotnode Properties

	The Time Plot node displays one or more sets of time series data. Typically, you would first use a Time Intervals node to create a <code>TimeLabel</code> field, which would be used to label the x axis.
---	---

Example

```
node = stream.create("timeplot", "My node")
node.setPropertyValue("y_fields", ["sales", "men", "women"])
node.setPropertyValue("panel", True)
node.setPropertyValue("normalize", True)
node.setPropertyValue("line", True)
node.setPropertyValue("smoother", True)
node.setPropertyValue("use_records_limit", True)
node.setPropertyValue("records_limit", 2000)
# Appearance settings
node.setPropertyValue("symbol_size", 2.0)
```

Table 1. timeplotnode properties

timeplotnode properties	Data type	Property description
plot_series	Series Models	
use_custom_x_field	flag	
x_field	field	
y_fields	list	
panel	flag	
normalize	flag	
line	flag	
points	flag	
point_type	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
smoother	flag	You can add smoothers to the plot only if you set <code>panel</code> to <code>True</code> .
use_records_limit	flag	
records_limit	integer	
symbol_size	number	Specifies a symbol size.
panel_layout	Horizontal Vertical	

Related information

- [Node properties overview](#)
- [Graph node common properties](#)
- [evaluationnode Properties](#)
- [graphboardnode Properties](#)
- [histogramnode Properties](#)
- [multiplotnode Properties](#)

- [plotnode Properties](#)
- [webnode Properties](#)

eplotnode Properties

	The E-Plot (Beta) node shows the relationship between numeric fields. It is similar to the Plot node, but its options differ and its output uses a new graphing interface specific to this node. Use the beta-level node to play around with new graphing features.
---	---

Table 1. eplotnode properties

eplotnode properties	Data type	Property description
x_field	<i>string</i>	Specify the field to display on the horizontal X axis.
y_field	<i>string</i>	Specify the field to display on the vertical Y axis.
color_field	<i>string</i>	Specify the field to use for the color map overlay in the output, if desired.
size_field	<i>string</i>	Specify the field to use for the size map overlay in the output, if desired.
shape_field	<i>string</i>	Specify the field to use for the shape map overlay in the output, if desired.
interested_fields	<i>string</i>	Specify the fields you'd like to include in the output.
records_limit	<i>integer</i>	Specify a number for the maximum number of records to plot in the output. 2000 is the default.
if_over_limit	<i>Boolean</i>	Specify whether to use the Sample option or the Use all data option if the records_limit is surpassed. Sample is the default, and it randomly samples the data until it hits the records_limit . If you specify Use all data to ignore the records_limit and plot all data points, note that this may dramatically decrease performance.

tsnenode Properties

	t-Distributed Stochastic Neighbor Embedding (t-SNE) is a tool for visualizing high-dimensional data. It converts affinities of data points to probabilities. This t-SNE node in SPSS® Modeler is implemented in Python and requires the scikit-learn® Python library.
---	--

Table 1. tsnenode properties

tsnenode properties	Data type	Property description
mode_type	<i>string</i>	Specify simple or expert mode.
n_components	<i>string</i>	Dimension of the embedded space (2D or 3D). Specify 2 or 3 . Default is 2 .
method	<i>string</i>	Specify barnes_hut or exact . Default is barnes_hut .
init	<i>string</i>	Initialization of embedding. Specify random or pca . Default is random .
target_field Renamed to target starting with version 18.2.1.1	<i>string</i>	Target field name. It will be a colormap on the output graph. The graph will use one color if no target field is specified.
perplexity	<i>float</i>	The perplexity is related to the number of nearest neighbors used in other manifold learning algorithms. Larger datasets usually require a larger perplexity. Consider selecting a value between 5 and 50. Default is 30 .
early_exaggeration	<i>float</i>	Controls how tight the natural clusters in the original space are in the embedded space, and how much space will be between them. Default is 12.0 .
learning_rate	<i>float</i>	Default is 200 .
n_iter	<i>integer</i>	Maximum number of iterations for the optimization. Set to at least 250 . Default is 1000 .
angle	<i>float</i>	The angular size of the distant node as measured from a point. Specify a value in the range of 0-1. Default is 0.5 .
enable_random_seed	<i>Boolean</i>	Set to true to enable the random_seed parameter. Default is false .
random_seed	<i>integer</i>	The random number seed to use. Default is None .
n_iter_without_progress	<i>integer</i>	Maximum iterations without progress. Default is 300 .

tsnenode properties	Data type	Property description
<code>min_grad_norm</code>	<code>string</code>	If the gradient norm is below this threshold, the optimization will be stopped. Default is <code>1.0E-7</code> . Possible values are: <ul style="list-style-type: none">• <code>1.0E-1</code>• <code>1.0E-2</code>• <code>1.0E-3</code>• <code>1.0E-4</code>• <code>1.0E-5</code>• <code>1.0E-6</code>• <code>1.0E-7</code>• <code>1.0E-8</code>
<code>isGridSearch</code>	<code>Boolean</code>	Set to <code>true</code> to perform t-SNE with several different perplexities. Default is <code>false</code> .
<code>output_Rename</code>	<code>Boolean</code>	Specify <code>true</code> if you want to provide a custom name, or <code>false</code> to name the output automatically. Default is <code>false</code> .
<code>output_to</code>	<code>string</code>	Specify <code>Screen</code> or <code>Output</code> . Default is <code>Screen</code> .
<code>full_filename</code>	<code>string</code>	Specify the output file name.
<code>output_file_type</code>	<code>string</code>	Output file format. Specify <code>HTML</code> or <code>Output object</code> . Default is <code>HTML</code> .

webnode Properties



The Web node illustrates the strength of the relationship between values of two or more symbolic (categorical) fields. The graph uses lines of various widths to indicate connection strength. You might use a Web node, for example, to explore the relationship between the purchase of a set of items at an e-commerce site.

Example

```
node = stream.create("web", "My node")
# "Plot" tab
node.setPropertyValue("use_directed_web", True)
node.setPropertyValue("to_field", "Drug")
node.setPropertyValue("fields", ["BP", "Cholesterol", "Sex", "Drug"])
node.setPropertyValue("from_fields", ["BP", "Cholesterol", "Sex"])
node.setPropertyValue("true_flags_only", False)
node.setPropertyValue("line_values", "Absolute")
node.setPropertyValue("strong_links_heavier", True)
# "Options" tab
node.setPropertyValue("max_num_links", 300)
node.setPropertyValue("links_above", 10)
node.setPropertyValue("num_links", "ShowAll")
node.setPropertyValue("discard_links_min", True)
node.setPropertyValue("links_min_records", 5)
node.setPropertyValue("discard_links_max", True)
node.setPropertyValue("weak_below", 10)
node.setPropertyValue("strong_above", 19)
node.setPropertyValue("link_size_continuous", True)
node.setPropertyValue("web_display", "Circular")
```

Table 1. webnode properties

webnode properties	Data type	Property description
<code>use_directed_web</code>	<code>flag</code>	
<code>fields</code>	<code>list</code>	
<code>to_field</code>	<code>field</code>	
<code>from_fields</code>	<code>list</code>	
<code>true_flags_only</code>	<code>flag</code>	
<code>line_values</code>	<code>Absolute OverallPct PctLarger PctSmaller</code>	
<code>strong_links_heavier</code>	<code>flag</code>	
<code>num_links</code>	<code>ShowMaximum ShowLinksAbove ShowAll</code>	
<code>max_num_links</code>	<code>number</code>	
<code>links_above</code>	<code>number</code>	
<code>discard_links_min</code>	<code>flag</code>	
<code>links_min_records</code>	<code>number</code>	
<code>discard_links_max</code>	<code>flag</code>	
<code>links_max_records</code>	<code>number</code>	

webnode properties	Data type	Property description
<code>weak_below</code>	<code>number</code>	
<code>strong_above</code>	<code>number</code>	
<code>link_size_continuos</code>	<code>flag</code>	
<code>web_display</code>	<code>Circular Network Directed Grid</code>	
<code>graph_background</code>	<code>color</code>	Standard graph colors are described at the beginning of this section.
<code>symbol_size</code>	<code>number</code>	Specifies a symbol size.

Related information

- [Node properties overview](#)
- [Graph node common properties](#)
- [evaluationnode Properties](#)
- [graphboardnode Properties](#)
- [histogramnode Properties](#)
- [multiplotnode Properties](#)
- [plotnode Properties](#)
- [timeplotnode Properties](#)

Modeling Node Properties

- [Common modeling node properties](#)
- [anomalydetectionnode properties](#)
- [apriorinode properties](#)
- [associationrulesnode properties](#)
- [autoclassifiernode properties](#)
- [autoclusternode properties](#)
- [autonumericnode properties](#)
- [bayesnetnode properties](#)
- [c50node properties](#)
- [carmanode properties](#)
- [cartnode properties](#)
- [chaidnode properties](#)
- [coxregnode properties](#)
- [decisionlistnode properties](#)
- [discriminantnode properties](#)
- [extensionmodelnode properties](#)
- [factornode properties](#)
- [featureselectionnode properties](#)
- [genlinnode properties](#)
- [glmmnode properties](#)
- [gle properties](#)
- [kmeansnode properties](#)
- [kmeansasnode properties](#)
- [knnnode properties](#)
- [kohonennode properties](#)
- [linearnode properties](#)
- [linearasnode properties](#)
- [logregnode properties](#)
- [lsvmnode properties](#)
- [neuralnetnode properties](#)
- [neuralnetworknode properties](#)
- [questnode properties](#)
- [randomtrees properties](#)
- [regressionnode properties](#)
- [sequencenode properties](#)
- [slrnnnode properties](#)
- [statisticsmodelnode properties](#)
- [stpnnode properties](#)
- [svmnode properties](#)
- [tcmnode Properties](#)
- [ts properties](#)

- [treeas properties](#)
- [twostepnode Properties](#)
- [twostepAS Properties](#)

Common modeling node properties

The following properties are common to some or all modeling nodes. Any exceptions are noted in the documentation for individual modeling nodes as appropriate.

Table 1. Common modeling node properties

Property	Values	Property description
<code>custom_fields</code>	<code>flag</code>	If true, allows you to specify target, input, and other fields for the current node. If false, the current settings from an upstream Type node are used.
<code>target or targets</code>	<code>field</code> or <code>[field1 ... fieldN]</code>	Specifies a single target field or multiple target fields depending on the model type.
<code>inputs</code>	<code>[field1 ... fieldN]</code>	Input or predictor fields used by the model.
<code>partition</code>	<code>field</code>	
<code>use_partitioned_data</code>	<code>flag</code>	If a partition field is defined, this option ensures that only data from the training partition is used to build the model.
<code>use_split_data</code>	<code>flag</code>	
<code>splits</code>	<code>[field1 ... fieldN]</code>	Specifies the field or fields to use for split modeling. Effective only if <code>use_split_data</code> is set to <code>True</code> .
<code>use_frequency</code>	<code>flag</code>	Weight and frequency fields are used by specific models as noted for each model type.
<code>frequency_field</code>	<code>field</code>	
<code>use_weight</code>	<code>flag</code>	
<code>weight_field</code>	<code>field</code>	
<code>use_model_name</code>	<code>flag</code>	
<code>model_name</code>	<code>string</code>	Custom name for new model.
<code>mode</code>	<code>Simple</code> <code>Expert</code>	

anomalydetectionnode properties



The Anomaly Detection node identifies unusual cases, or outliers, that do not conform to patterns of “normal” data. With this node, it is possible to identify outliers even if they do not fit any previously known patterns and even if you are not exactly sure what you are looking for.

Example

```
node = stream.create("anomalydetection", "My node")
node.setPropertyValue("anomaly_method", "PerRecords")
node.setPropertyValue("percent_records", 95)
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("peer_group_num_auto", True)
node.setPropertyValue("min_num_peer_groups", 3)
node.setPropertyValue("max_num_peer_groups", 10)
```

Table 1. anomalydetectionnode properties

anomalydetectionnode Properties	Values	Property description
<code>inputs</code>	<code>[field1 ... fieldN]</code>	Anomaly Detection models screen records based on the specified input fields. They do not use a target field. Weight and frequency fields are also not used. See the topic Common modeling node properties for more information.
<code>mode</code>	<code>Expert</code> <code>Simple</code>	
<code>anomaly_method</code>	<code>IndexLevel</code> <code>PerRecords</code> <code>NumRecords</code>	Specifies the method used to determine the cutoff value for flagging records as anomalous.
<code>index_level</code>	<code>number</code>	Specifies the minimum cutoff value for flagging anomalies.
<code>percent_records</code>	<code>number</code>	Sets the threshold for flagging records based on the percentage of records in the training data.
<code>num_records</code>	<code>number</code>	Sets the threshold for flagging records based on the number of records in the training data.
<code>num_fields</code>	<code>integer</code>	The number of fields to report for each anomalous record.

anomalydetect ionnode Properties	Values	Property description
impute_missing_values	<i>flag</i>	
adjustment_coeff	<i>number</i>	Value used to balance the relative weight given to continuous and categorical fields in calculating the distance.
peer_group_num_auto	<i>flag</i>	Automatically calculates the number of peer groups.
min_num_peer_groups	<i>integer</i>	Specifies the minimum number of peer groups used when peer_group_num_auto is set to True .
max_num_per_groups	<i>integer</i>	Specifies the maximum number of peer groups.
num_peer_groups	<i>integer</i>	Specifies the number of peer groups used when peer_group_num_auto is set to False .
noise_level	<i>number</i>	Determines how outliers are treated during clustering. Specify a value between 0 and 0.5.
noise_ratio	<i>number</i>	Specifies the portion of memory allocated for the component that should be used for noise buffering. Specify a value between 0 and 0.5.

apriorinode properties



The Apriori node extracts a set of rules from the data, pulling out the rules with the highest information content. Apriori offers five different methods of selecting rules and uses a sophisticated indexing scheme to process large data sets efficiently. For large problems, Apriori is generally faster to train; it has no arbitrary limit on the number of rules that can be retained, and it can handle rules with up to 32 preconditions. Apriori requires that input and output fields all be categorical but delivers better performance because it is optimized for this type of data.

Example

```
node = stream.create("apriori", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("partition", "Test")
# For non-transactional
node.setPropertyValue("use_transactional_data", False)
node.setPropertyValue("consequents", ["Age"])
node.setPropertyValue("antecedents", ["BP", "Cholesterol", "Drug"])
# For transactional
node.setPropertyValue("use_transactional_data", True)
node.setPropertyValue("id_field", "Age")
node.setPropertyValue("contiguous", True)
node.setPropertyValue("content_field", "Drug")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Apriori_bp_choles_drug")
node.setPropertyValue("min_supp", 7.0)
node.setPropertyValue("min_conf", 30.0)
node.setPropertyValue("max_antecedents", 7)
node.setPropertyValue("true_flags", False)
node.setPropertyValue("optimize", "Memory")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("evaluation", "ConfidenceRatio")
node.setPropertyValue("lower_bound", 7)
```

Table 1. apriorinode properties

apriorinode Properties	Values	Property description
consequents	<i>field</i>	Apriori models use Consequents and Antecedents in place of the standard target and input fields. Weight and frequency fields are not used. See the topic Common modeling node properties for more information.
antecedents	<i>[field1 ... fieldN]</i>	
min_supp	<i>number</i>	
min_conf	<i>number</i>	
max_antecedents	<i>number</i>	
true_flags	<i>flag</i>	
optimize	<i>Speed Memory</i>	

apriorinode Properties	Values	Property description
use_transactional_data	<i>flag</i>	When the value is true , the score for each transaction ID is independent from other transaction IDs. When the data to be scored is too large to obtain acceptable performance, we recommend separating the data.
contiguous	<i>flag</i>	
id_field	<i>string</i>	
content_field	<i>string</i>	
mode	Simple Expert	
evaluation	RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare	
lower_bound	<i>number</i>	
optimize	Speed Memory	Use to specify whether model building should be optimized for speed or for memory.

associationrulesnode properties



The Association Rules Node is similar to the Apriori Node; however, unlike Apriori, the Association Rules Node can process list data. In addition, the Association Rules Node can be used with IBM® SPSS® Analytic Server to process big data and take advantage of faster parallel processing.

Table 1. associationrulesnode properties

associationrulesnode properties	Data type	Property description
predictions	<i>field</i>	Fields in this list can only appear as a predictor of a rule
conditions	<i>[field1...fieldN]</i>	Fields in this list can only appear as a condition of a rule
max_rule_conditions	<i>integer</i>	The maximum number of conditions that can be included in a single rule. Minimum 1, maximum 9.
max_rule_predictions	<i>integer</i>	The maximum number of predictions that can be included in a single rule. Minimum 1, maximum 5.
max_num_rules	<i>integer</i>	The maximum number of rules that can be considered as part of rule building. Minimum 1, maximum 10,000.
rule_criterion_top_n	Confidence Rulesupport Lift Conditionsupport Deployability	The rule criterion that determines the value by which the top "N" rules in the model are chosen.
true_flags	<i>Boolean</i>	Setting as Y determines that only the true values for flag fields are considered during rule building.
rule_criterion	<i>Boolean</i>	Setting as Y determines that the rule criterion values are used for excluding rules during model building.
min_confidence	<i>number</i>	0.1 to 100 - the percentage value for the minimum required confidence level for a rule produced by the model. If the model produces a rule with a confidence level less than the value specified here the rule is discarded.
min_rule_support	<i>number</i>	0.1 to 100 - the percentage value for the minimum required rule support for a rule produced by the model. If the model produces a rule with a rule support level less than the specified value the rule is discarded.
min_condition_support	<i>number</i>	0.1 to 100 - the percentage value for the minimum required condition support for a rule produced by the model. If the model produces a rule with a condition support level less than the specified value the rule is discarded.
min_lift	<i>integer</i>	1 to 10 - represents the minimum required lift for a rule produced by the model. If the model produces a rule with a lift level less than the specified value the rule is discarded.
exclude_rules	<i>Boolean</i>	Used to select a list of related fields from which you do not want the model to create rules. Example: set :gsarsnode.exclude_rules = [[[field1,field2, field3]],[[field4, field5]]] - where each list of fields separated by [] is a row in the table.
num_bins	<i>integer</i>	Set the number of automatic bins that continuous fields are binned to. Minimum 2, maximum 10.
max_list_length	<i>integer</i>	Applies to any list fields for which the maximum length is not known. Elements in the list up until the number specified here are included in the model build; any further elements are discarded. Minimum 1, maximum 100.

associationrulesnode properties	Data type	Property description
output_confidence	Boolean	
output_rule_support	Boolean	
output_lift	Boolean	
output_condition_support	Boolean	
output_deployability	Boolean	
rules_to_display	upto all	The maximum number of rules to display in the output tables.
display_upto	integer	If upto is set in rules_to_display, set the number of rules to display in the output tables. Minimum 1.
field_transformations	Boolean	
records_summary	Boolean	
rule_statistics	Boolean	
most_frequent_values	Boolean	
most_frequent_fields	Boolean	
word_cloud	Boolean	
word_cloud_sort	Confidence Rulesupport Lift Conditionsupport Deployability	
word_cloud_display	integer	Minimum 1, maximum 20
max_predictions	integer	The maximum number of rules that can be applied to each input to the score.
criterion	Confidence Rulesupport Lift Conditionsupport Deployability	Select the measure used to determine the strength of rules.
allow_repeats	Boolean	Determine whether rules with the same prediction are included in the score.
check_input	NoPredictions Predictions NoCheck	

autoclassifiernode properties



The Auto Classifier node creates and compares a number of different models for binary outcomes (yes or no, churn or do not churn, and so on), allowing you to choose the best approach for a given analysis. A number of modeling algorithms are supported, making it possible to select the methods you want to use, the specific options for each, and the criteria for comparing the results. The node generates a set of models based on the specified options and ranks the best candidates according to the criteria you specify.

Example

```
node = stream.create("autoclassifier", "My node")
node.setPropertyValue("ranking_measure", "Accuracy")
node.setPropertyValue("ranking_dataset", "Training")
node.setPropertyValue("enable_accuracy_limit", True)
node.setPropertyValue("accuracy_limit", 0.9)
node.setPropertyValue("calculate_variable_importance", True)
node.setPropertyValue("use_costs", True)
node.setPropertyValue("svm", False)
```

Table 1. autoclassifiernode properties

autoclassifiernode Properties	Values	Property description
target	field	For flag targets, the Auto Classifier node requires a single target and one or more input fields. Weight and frequency fields can also be specified. See the topic Common modeling node properties for more information.

autoclassifiernode Properties	Values	Property description
ranking_measure	Accuracy Area_under_curve Profit Lift Num variables	
ranking_dataset	Training Test	
number_of_models	integer	Number of models to include in the model nugget. Specify an integer between 1 and 100.
calculate_variable_importance	flag	
enable_accuracy_limit	flag	
accuracy_limit	integer	Integer between 0 and 100.
enable_area_under_curve_limit	flag	
area_under_curve_limit	number	Real number between 0.0 and 1.0.
enable_profit_limit	flag	
profit_limit	number	Integer greater than 0.
enable_lift_limit	flag	
lift_limit	number	Real number greater than 1.0.
enable_number_of_variables_limit	flag	
number_of_variables_limit	number	Integer greater than 0.
use_fixed_cost	flag	
fixed_cost	number	Real number greater than 0.0.
variable_cost	field	
use_fixed_revenue	flag	
fixed_revenue	number	Real number greater than 0.0.
variable_revenue	field	
use_fixed_weight	flag	
fixed_weight	number	Real number greater than 0.0
variable_weight	field	
lift_percentile	number	Integer between 0 and 100.
enable_model_build_time_limit	flag	
model_build_time_limit	number	Integer set to the number of minutes to limit the time taken to build each individual model.
enable_stop_after_time_limit	flag	
stop_after_time_limit	number	Real number set to the number of hours to limit the overall elapsed time for an auto classifier run.
enable_stop_after_valid_model_produced	flag	
use_costs	flag	
<algorithm>	flag	Enables or disables the use of a specific algorithm.
<algorithm>. <property>	string	Sets a property value for a specific algorithm. See the topic Setting Algorithm Properties for more information.

- [Setting Algorithm Properties](#)

Setting Algorithm Properties

For the Auto Classifier, Auto Numeric, and Auto Cluster nodes, properties for specific algorithms used by the node can be set using the general form:

```
autonode.setKeyedPropertyValue(<algorithm>, <property>, <value>)
```

For example:

```
node.setKeyedPropertyValue("neuralnetwork", "method", "MultilayerPerceptron")
```

Algorithm names for the Auto Classifier node are `cart`, `chaid`, `quest`, `c50`, `logreg`, `decisionlist`, `bayesnet`, `discriminant`, `svm` and `knn`.

Algorithm names for the Auto Numeric node are `cart`, `chaid`, `neuralnetwork`, `genlin`, `svm`, `regression`, `linear` and `knn`.

Algorithm names for the Auto Cluster node are `twostep`, `k-means`, and `kohonen`.

Property names are standard as documented for each algorithm node.

Algorithm properties that contain periods or other punctuation must be wrapped in single quotes, for example:

```
node.setKeyedPropertyValue("logreg", "tolerance", "1.0E-5")
```

Multiple values can also be assigned for property, for example:

```
node.setKeyedPropertyValue("decisionlist", "search_direction", ["Up", "Down"])
```

To enable or disable the use of a specific algorithm:

```
node.setPropertyValue("chaid", True)
```

Note: In cases where certain algorithm options are not available in the Auto Classifier node, or when only a single value can be specified rather than a range of values, the same limits apply with scripting as when accessing the node in the standard manner.

Related information

- [autoclassifiernode properties](#)
 - [autoclusternode properties](#)
 - [autonumericnode properties](#)
-

autoclusternode properties



The Auto Cluster node estimates and compares clustering models, which identify groups of records that have similar characteristics. The node works in the same manner as other automated modeling nodes, allowing you to experiment with multiple combinations of options in a single modeling pass. Models can be compared using basic measures with which to attempt to filter and rank the usefulness of the cluster models, and provide a measure based on the importance of particular fields.

Example

```
node = stream.create("autocluster", "My node")
node.setPropertyValue("ranking_measure", "Silhouette")
node.setPropertyValue("ranking_dataset", "Training")
node.setPropertyValue("enable_silhouette_limit", True)
node.setPropertyValue("silhouette_limit", 5)
```

Table 1. autoclusternode properties

autoclusternode Properties	Values	Property description
<code>evaluation</code>	<code>field</code>	Note: Auto Cluster node only. Identifies the field for which an importance value will be calculated. Alternatively, can be used to identify how well the cluster differentiates the value of this field and, therefore; how well the model will predict this field.
<code>ranking_measure</code>	<code>Silhouette</code> <code>Num_clusters</code> <code>Size_smallest_cluster</code> <code>Size_largest_cluster</code> <code>Smallest_to_largest_Importance</code>	
<code>ranking_dataset</code>	<code>Training</code> <code>Test</code>	
<code>summary_limit</code>	<code>integer</code>	Number of models to list in the report. Specify an integer between 1 and 100.
<code>enable_silhouette_limit</code>	<code>flag</code>	
<code>silhouette_limit</code>	<code>integer</code>	Integer between 0 and 100.
<code>enable_numberless_limit</code>	<code>flag</code>	

autoclusternode Properties	Values	Property description
number_less_limit	number	Real number between 0.0 and 1.0.
enable_number_greater_limit	flag	
number_greater_limit	number	Integer greater than 0.
enable_smallest_cluster_limit	flag	
smallest_cluster_units	Percentage Counts	
smallest_cluster_limit_percentage	number	
smallest_cluster_limit_count	integer	Integer greater than 0.
enable_largest_cluster_limit	flag	
largest_cluster_units	Percentage Counts	
largest_cluster_limit_percentage	number	
largest_cluster_limit_count	integer	
enable_smallest_largest_limit	flag	
smallest_largest_limit	number	
enable_importance_limit	flag	
importance_limit_condition	Greater_than Less_than	
importance_limit_greater_than	number	Integer between 0 and 100.
importance_limit_less_than	number	Integer between 0 and 100.
<algorithm>	flag	Enables or disables the use of a specific algorithm.
<algorithm>. <property>	string	Sets a property value for a specific algorithm. See the topic Setting Algorithm Properties for more information.

autonumericnode properties



The Auto Numeric node estimates and compares models for continuous numeric range outcomes using a number of different methods. The node works in the same manner as the Auto Classifier node, allowing you to choose the algorithms to use and to experiment with multiple combinations of options in a single modeling pass. Supported algorithms include neural networks, C&R Tree, CHAID, linear regression, generalized linear regression, and support vector machines (SVM). Models can be compared based on correlation, relative error, or number of variables used.

Example

```
node = stream.create("autonumeric", "My node")
node.setPropertyValue("ranking_measure", "Correlation")
node.setPropertyValue("ranking_dataset", "Training")
node.setPropertyValue("enable_correlation_limit", True)
node.setPropertyValue("correlation_limit", 0.8)
node.setPropertyValue("calculate_variable_importance", True)
node.setPropertyValue("neuralnetwork", True)
node.setPropertyValue("chaid", False)
```

Table 1. autonumericnode properties

autonumericnode Properties	Values	Property description
custom_fields	<i>flag</i>	If True, custom field settings will be used instead of type node settings.
target	<i>field</i>	The Auto Numeric node requires a single target and one or more input fields. Weight and frequency fields can also be specified. See the topic Common modeling node properties for more information.
inputs	<i>[field1 ... field2]</i>	
partition	<i>field</i>	
use_frequency	<i>flag</i>	
frequency_field	<i>field</i>	
use_weight	<i>flag</i>	
weight_field	<i>field</i>	
use_partitioned_data	<i>flag</i>	If a partition field is defined, only the training data are used for model building.
ranking_measure	<i>Correlation NumberOfFields</i>	
ranking_dataset	Test Training	
number_of_models	<i>integer</i>	Number of models to include in the model nugget. Specify an integer between 1 and 100.
calculate_variable_importance	<i>flag</i>	
enable_correlation_limit	<i>flag</i>	
correlation_limit	<i>integer</i>	
enable_number_of_fields_limit	<i>flag</i>	
number_of_fields_limit	<i>integer</i>	
enable_relative_error_limit	<i>flag</i>	
relative_error_limit	<i>integer</i>	
enable_model_build_time_limit	<i>flag</i>	
model_build_time_limit	<i>integer</i>	
enable_stop_after_time_limit	<i>flag</i>	
stop_after_time_limit	<i>integer</i>	
stop_if_valid_model	<i>flag</i>	
<algorithm>	<i>flag</i>	Enables or disables the use of a specific algorithm.
<algorithm>. <property>	<i>string</i>	Sets a property value for a specific algorithm. See the topic Setting Algorithm Properties for more information.

bayesnetnode properties



The Bayesian Network node enables you to build a probability model by combining observed and recorded evidence with real-world knowledge to establish the likelihood of occurrences. The node focuses on Tree Augmented Naïve Bayes (TAN) and Markov Blanket networks that are primarily used for classification.

Example

```
node = stream.create("bayesnet", "My node")
node.setPropertyValue("continue_training_existing_model", True)
node.setPropertyValue("structure_type", "MarkovBlanket")
node.setPropertyValue("use_feature_selection", True)
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("independence", "Pearson")
```

Table 1. bayesnetnode properties

bayesnetnode Properties	Values	Property description
inputs	<i>[field1 ... fieldN]</i>	Bayesian network models use a single target field, and one or more input fields. Continuous fields are automatically binned. See the topic Common modeling node properties for more information.
continue_training_existing_model	<i>flag</i>	
structure_type	TAN MarkovBlanket	Select the structure to be used when building the Bayesian network.
use_feature_selection	<i>flag</i>	
parameter_learning_method	Likelihood Bayes	Specifies the method used to estimate the conditional probability tables between nodes where the values of the parents are known.
mode	Expert Simple	
missing_values	<i>flag</i>	
all_probabilities	<i>flag</i>	
independence	Likelihood Pearson	Specifies the method used to determine whether paired observations on two variables are independent of each other.
significance_level	<i>number</i>	Specifies the cutoff value for determining independence.
maximal_conditioning_set	<i>number</i>	Sets the maximal number of conditioning variables to be used for independence testing.
inputs_always_selected	<i>[field1 ... fieldN]</i>	Specifies which fields from the dataset are always to be used when building the Bayesian network. Note: The target field is always selected.
maximum_number_inputs	<i>number</i>	Specifies the maximum number of input fields to be used in building the Bayesian network.
calculate_variable_importance	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

c50node properties



The C5.0 node builds either a decision tree or a rule set. The model works by splitting the sample based on the field that provides the maximum information gain at each level. The target field must be categorical. Multiple splits into more than two subgroups are allowed.

Example

```
node = stream.create("c50", "My node")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "C5_Drug")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("output_type", "DecisionTree")
node.setPropertyValue("use_xval", True)
node.setPropertyValue("xval_num_folds", 3)
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("favor", "Generality")
node.setPropertyValue("min_child_records", 3)
# "Costs" tab
node.setPropertyValue("use_costs", True)
node.setPropertyValue("costs", [{"drugA": "drugX", 2}])
```

Table 1. c50node properties

c50node Properties	Values	Property description
target	<i>field</i>	C50 models use a single target field and one or more input fields. A weight field can also be specified. See the topic Common modeling node properties for more information.
output_type	DecisionTree RuleSet	
group_symbolics	<i>flag</i>	
use_boost	<i>flag</i>	
boost_num_trials	<i>number</i>	

c50node Properties	Values	Property description
use_xval	flag	
xval_num_folds	number	
mode	Simple Expert	
favor	Accuracy Generality	Favor accuracy or generality.
expected_noise	number	
min_child_records	number	
pruning_severity	number	
use_costs	flag	
costs	structured	This is a structured property. See the example for usage.
use_winning	flag	
use_global_pruning	flag	On (True) by default.
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

carmanode properties



The CARMA model extracts a set of rules from the data without requiring you to specify input or target fields. In contrast to Apriori the CARMA node offers build settings for rule support (support for both antecedent and consequent) rather than just antecedent support. This means that the rules generated can be used for a wider variety of applications—for example, to find a list of products or services (antecedents) whose consequent is the item that you want to promote this holiday season.

Example

```
node = stream.create("carma", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("use_transactional_data", True)
node.setPropertyValue("inputs", ["BP", "Cholesterol", "Drug"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "age_bp_drug")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("min_supp", 10.0)
node.setPropertyValue("min_conf", 30.0)
node.setPropertyValue("max_size", 5)
# Expert Options
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("use_pruning", True)
node.setPropertyValue("pruning_value", 300)
node.setPropertyValue("vary_support", True)
node.setPropertyValue("estimated_transactions", 30)
node.setPropertyValue("rules_without_antecedents", True)
```

Table 1. carmanode properties

carmanode Properties	Values	Property description
inputs	[field1 ... fieldn]	CARMA models use a list of input fields, but no target. Weight and frequency fields are not used. See the topic Common modeling node properties for more information.
id_field	field	Field used as the ID field for model building.
contiguous	flag	Used to specify whether IDs in the ID field are contiguous.
use_transactional_data	flag	
content_field	field	
min_supp	number(percent)	Relates to rule support rather than antecedent support. The default is 20%.
min_conf	number(percent)	The default is 20%.
max_size	number	The default is 10.
mode	Simple Expert	The default is Simple.

cartnode Properties	Values	Property description
<code>exclude_multiple</code>	<code>flag</code>	Excludes rules with multiple consequents. The default is <code>False</code> .
<code>use_pruning</code>	<code>flag</code>	The default is <code>False</code> .
<code>pruning_value</code>	<code>number</code>	The default is 500.
<code>vary_support</code>	<code>flag</code>	
<code>estimated_transactions</code>	<code>integer</code>	
<code>rules_without_ancestors</code>	<code>flag</code>	

cartnode properties

	The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered “pure” if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).
---	--

Example

```
node = stream.createAt("cart", "My node", 200, 100)
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "BP", "Cholesterol"])
# "Build Options" tab, "Objective" panel
node.setPropertyValue("model_output_type", "InteractiveBuilder")
node.setPropertyValue("use_tree_directives", True)
node.setPropertyValue("tree_directives", """Grow Node Index 0 Children 1 2
Grow Node Index 2 Children 3 4""")
# "Build Options" tab, "Basics" panel
node.setPropertyValue("prune_tree", False)
node.setPropertyValue("use_std_err_rule", True)
node.setPropertyValue("std_err_multiplier", 3.0)
node.setPropertyValue("max_surrogates", 7)
# "Build Options" tab, "Stopping Rules" panel
node.setPropertyValue("use_percentage", True)
node.setPropertyValue("min_parent_records_pc", 5)
node.setPropertyValue("min_child_records_pc", 3)
# "Build Options" tab, "Advanced" panel
node.setPropertyValue("min_impurity", 0.0003)
node.setPropertyValue("impurity_measure", "Twoing")
# "Model Options" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Cart_Drug")
```

Table 1. cartnode properties

cartnode Properties	Values	Property description
<code>target</code>	<code>field</code>	C&R Tree models require a single target and one or more input fields. A frequency field can also be specified. See the topic Common modeling node properties for more information.
<code>continue_training_existing_model</code>	<code>flag</code>	
<code>objective</code>	<code>StandardBoosting</code> <code>Bagging psm</code>	<code>psm</code> is used for very large datasets, and requires a Server connection.
<code>model_output_type</code>	<code>Single</code> <code>InteractiveBuilder</code>	
<code>use_tree_directives</code>	<code>flag</code>	
<code>tree_directives</code>	<code>string</code>	Specify directives for growing the tree. Directives can be wrapped in triple quotes to avoid escaping newlines or quotes. Note that directives may be highly sensitive to minor changes in data or modeling options and may not generalize to other datasets. See the example for usage.
<code>use_max_depth</code>	<code>Default</code> <code>Custom</code>	
<code>max_depth</code>	<code>integer</code>	Maximum tree depth, from 0 to 1000. Used only if <code>use_max_depth = Custom</code> .

cartnode Properties	Values	Property description
<code>prune_tree</code>	<code>flag</code>	Prune tree to avoid overfitting.
<code>use_std_err</code>	<code>flag</code>	Use maximum difference in risk (in Standard Errors).
<code>std_err_mult_iplier</code>	<code>number</code>	Maximum difference.
<code>max_surrogates</code>	<code>number</code>	Maximum surrogates.
<code>use_percentage</code>	<code>flag</code>	
<code>min_parent_records_pc</code>	<code>number</code>	
<code>min_child_records_pc</code>	<code>number</code>	
<code>min_parent_records_abs</code>	<code>number</code>	
<code>min_child_records_abs</code>	<code>number</code>	
<code>use_costs</code>	<code>flag</code>	
<code>costs</code>	<code>structured</code>	Structured property.
<code>priors</code>	<code>Data Equal</code> <code>Custom</code>	
<code>custom_priors</code>	<code>structured</code>	Structured property.
<code>adjust_priors</code>	<code>flag</code>	
<code>trails</code>	<code>number</code>	Number of component models for boosting or bagging.
<code>set_ensemble_method</code>	<code>Voting</code> <code>HighestProbability</code> <code>HighestMeanProbability</code>	Default combining rule for categorical targets.
<code>range_ensemble_method</code>	<code>Mean</code> <code>Median</code>	Default combining rule for continuous targets.
<code>large_boost</code>	<code>flag</code>	Apply boosting to very large data sets.
<code>min_impurity</code>	<code>number</code>	
<code>impurity_measure</code>	<code>Gini</code> <code>Twoing</code> <code>Ordered</code>	
<code>train_pct</code>	<code>number</code>	Overfit prevention set.
<code>set_random_seed</code>	<code>flag</code>	Replicate results option.
<code>seed</code>	<code>number</code>	
<code>calculate_variable_importance</code>	<code>flag</code>	
<code>calculate_raw_propensities</code>	<code>flag</code>	
<code>calculate_adjusted_propensities</code>	<code>flag</code>	
<code>adjusted_propensity_partition</code>	<code>Test Validation</code>	

chaidnode properties

	The CHAID node generates decision trees using chi-square statistics to identify optimal splits. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.
---	--

Example

```

filenode = stream.createAt("variablefile", "My node", 100, 100)
filenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node = stream.createAt("chaid", "My node", 200, 100)
stream.link(filenode, node)

node.setPropertyValue("custom_fields", True)

```

```

node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "CHAID")
node.setPropertyValue("method", "Chaid")
node.setPropertyValue("model_output_type", "InteractiveBuilder")
node.setPropertyValue("use_tree_directives", True)
node.setPropertyValue("tree_directives", "Test")
node.setPropertyValue("split_alpha", 0.03)
node.setPropertyValue("merge_alpha", 0.04)
node.setPropertyValue("chi_square", "Pearson")
node.setPropertyValue("use_percentage", False)
node.setPropertyValue("min_parent_records_abs", 40)
node.setPropertyValue("min_child_records_abs", 30)
node.setPropertyValue("epsilon", 0.003)
node.setPropertyValue("max_iterations", 75)
node.setPropertyValue("split_merged_categories", True)
node.setPropertyValue("bonferroni_adjustment", True)

```

Table 1. chaidnode properties

chaidnode Properties	Values	Property description
<code>target</code>	<code>field</code>	CHAID models require a single target and one or more input fields. A frequency field can also be specified. See the topic Common modeling node properties for more information.
<code>continue_training_existing_model</code>	<code>flag</code>	
<code>objective</code>	<code>Standard Boosting</code> <code>Bagging psm</code>	<code>psm</code> is used for very large datasets, and requires a Server connection.
<code>model_output_type</code>	<code>Single</code> <code>InteractiveBuilder</code>	
<code>use_tree_directives</code>	<code>flag</code>	
<code>tree_directives</code>	<code>string</code>	
<code>method</code>	<code>Chaid</code> <code>ExhaustiveChaid</code>	
<code>use_max_depth</code>	<code>Default</code> <code>Custom</code>	
<code>max_depth</code>	<code>integer</code>	Maximum tree depth, from 0 to 1000. Used only if <code>use_max_depth = Custom</code> .
<code>use_percentage</code>	<code>flag</code>	
<code>min_parent_records_pc</code>	<code>number</code>	
<code>min_child_records_pc</code>	<code>number</code>	
<code>min_parent_records_abs</code>	<code>number</code>	
<code>min_child_records_abs</code>	<code>number</code>	
<code>use_costs</code>	<code>flag</code>	
<code>costs</code>	<code>structured</code>	Structured property.
<code>trails</code>	<code>number</code>	Number of component models for boosting or bagging.
<code>set_ensemble_method</code>	<code>Voting</code> <code>HighestProbability</code> <code>HighestMeanProbability</code>	Default combining rule for categorical targets.
<code>range_ensemble_method</code>	<code>Mean</code> <code>Median</code>	Default combining rule for continuous targets.
<code>large_boost</code>	<code>flag</code>	Apply boosting to very large data sets.
<code>split_alpha</code>	<code>number</code>	Significance level for splitting.
<code>merge_alpha</code>	<code>number</code>	Significance level for merging.
<code>bonferroni_adjustment</code>	<code>flag</code>	Adjust significance values using Bonferroni method.
<code>split_merged_categories</code>	<code>flag</code>	Allow resplitting of merged categories.
<code>chi_square</code>	<code>Pearson</code> <code>LR</code>	Method used to calculate the chi-square statistic: Pearson or Likelihood Ratio
<code>epsilon</code>	<code>number</code>	Minimum change in expected cell frequencies..
<code>max_iterations</code>	<code>number</code>	Maximum iterations for convergence.
<code>set_random_seed</code>	<code>integer</code>	
<code>seed</code>	<code>number</code>	
<code>calculate_variable_importance</code>	<code>flag</code>	

chaidnode Properties	Values	Property description
calculate_raw_p	flag	
ropensities	flag	
calculate_adjusted_propensities	flag	
adjusted_proximity_partition	Test Validation	
maximum_number_of_models	integer	

coxregnode properties

	The Cox regression node enables you to build a survival model for time-to-event data in the presence of censored records. The model produces a survival function that predicts the probability that the event of interest has occurred at a given time (t) for given values of the input variables.
---	---

Example

```
node = stream.create("coxreg", "My node")
node.setPropertyValue("survival_time", "tenure")
node.setPropertyValue("method", "BackwardsStepwise")
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("removal_criterion", "Conditional")
node.setPropertyValue("survival", True)
```

Table 1. coxregnode properties

coxregnode Properties	Values	Property description
survival_time	field	Cox regression models require a single field containing the survival times.
target	field	Cox regression models require a single target field, and one or more input fields. See the topic Common modeling node properties for more information.
method	Enter Stepwise BackwardsStepwise	
groups	field	
model_type	MainEffects Custom	
custom_terms	["BP*Sex" "BP*Age"]	
mode	Expert Simple	
max_iterations	number	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
removal_criterion	LR Wald Conditional	
probability_entry	number	
probability_removal	number	
output_display	EachStep LastStep	
ci_enable	flag	
ci_value	90 95 99	
correlation	flag	
display_base_line	flag	
survival	flag	
hazard	flag	
log_minus_log	flag	
one_minus_survival	flag	

coxregnode Properties	Values	Property description
separate_line	<i>field</i>	
value	<i>number or string</i>	If no value is specified for a field, the default option "Mean" will be used for that field.

decisionlistnode properties

	The Decision List node identifies subgroups, or segments, that show a higher or lower likelihood of a given binary outcome relative to the overall population. For example, you might look for customers who are unlikely to churn or are most likely to respond favorably to a campaign. You can incorporate your business knowledge into the model by adding your own custom segments and previewing alternative models side by side to compare the results. Decision List models consist of a list of rules in which each rule has a condition and an outcome. Rules are applied in order, and the first rule that matches determines the outcome.
---	---

Example

```
node = stream.create("decisionlist", "My node")
node.setPropertyValue("search_direction", "Down")
node.setPropertyValue("target_value", 1)
node.setPropertyValue("max_rules", 4)
node.setPropertyValue("min_group_size_pct", 15)
```

Table 1. decisionlistnode properties

decisionlistnode Properties	Values	Property description
target	<i>field</i>	Decision List models use a single target and one or more input fields. A frequency field can also be specified. See the topic Common modeling node properties for more information.
model_output_type	<i>Model InteractiveBuilder</i>	
search_direction	<i>Up Down</i>	Relates to finding segments; where Up is the equivalent of High Probability, and Down is the equivalent of Low Probability..
target_value	<i>string</i>	If not specified, will assume true value for flags.
max_rules	<i>integer</i>	The maximum number of segments excluding the remainder.
min_group_size	<i>integer</i>	Minimum segment size.
min_group_size_pct	<i>number</i>	Minimum segment size as a percentage.
confidence_level	<i>number</i>	Minimum threshold that an input field has to improve the likelihood of response (give lift), to make it worth adding to a segment definition.
max_segments_per_rule	<i>integer</i>	
mode	<i>Simple Expert</i>	
bin_method	<i>EqualWidth EqualCount</i>	
bin_count	<i>number</i>	
max_models_per_cycle	<i>integer</i>	Search width for lists.
max_rules_per_cycle	<i>integer</i>	Search width for segment rules.
segment_growth	<i>number</i>	
include_missing	<i>flag</i>	
final_results_only	<i>flag</i>	
reuse_fields	<i>flag</i>	Allows attributes (input fields which appear in rules) to be re-used.
max_alternatives	<i>integer</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	<i>Test Validation</i>	

discriminantnode properties

	Discriminant analysis makes more stringent assumptions than logistic regression but can be a valuable alternative or supplement to a logistic regression analysis when those assumptions are met.
---	---

Example

```
node = stream.create("discriminant", "My node")
node.setPropertyValue("target", "custcat")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("method", "Stepwise")
```

Table 1. discriminantnode properties

discriminantnode Properties	Values	Property description
target	field	Discriminant models require a single target field and one or more input fields. Weight and frequency fields are not used. See the topic Common modeling node properties for more information.
method	Enter Stepwise	
mode	Simple Expert	
prior_probabilities	AllEqual ComputeFromSizes	
covariance_matrix	WithinGroups SeparateGroups	
means	flag	Statistics options in the Advanced Output dialog box.
univariate_anovas	flag	
box_m	flag	
within_group_covariance	flag	
within_groups_correlation	flag	
separate_groups_covariance	flag	
total_covariance	flag	
fishers	flag	
unstandardized	flag	
casewise_results	flag	Classification options in the Advanced Output dialog box.
limit_to_first	number	Default value is 10.
summary_table	flag	
leave_one_classification	flag	
combined_groups	flag	
separate_groups_covariance	flag	Matrices option Separate-groups covariance.
territorial_map	flag	
combined_groups	flag	Plot option Combined-groups.
separate_groups	flag	Plot option Separate-groups.
summary_of_steps	flag	
F_pairwise	flag	
stepwise_method	WilksLambda UnexplainedVariance MahalanobisDistance SmallestF RaosV	
V_to_enter	number	
criteria	UseValue UseProbability	
F_value_entry	number	Default value is 3.84.
F_value_removal	number	Default value is 2.71.
probability_entry	number	Default value is 0.05.
probability_removal	number	Default value is 0.10.

discriminantnode Properties	Values	Property description
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted propensity_partition	TestValidation	

extensionmodelnode properties



With the Extension Model node, you can run R or Python for spark scripts to build and score results.

Python for Spark example

```
##### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create('extension_build', "extension_build")
node.setPropertyValue("syntax_type", "Python")

build_script = """
import json
import spss.pyspark.runtime
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.linalg import DenseVector
from pyspark.mllib.tree import DecisionTree

ctx = spss.pyspark.runtime.getContext()
df = ctx.getSparkInputData()
schema = df.dtypes[:]

target = "Drug"
predictors = ["Age", "BP", "Sex", "Cholesterol", "Na", "K"]

def metaMap(row,schema):
    col = 0
    meta = []
    for (cname, ctype) in schema:
        if ctype == 'string':
            meta.append(set([row[col]]))
        else:
            meta.append((row[col],row[col]))
    col += 1
    return meta

def metaReduce(meta1,meta2,schema):
    col = 0
    meta = []
    for (cname, ctype) in schema:
        if ctype == 'string':
            meta.append(meta1[col].union(meta2[col]))
        else:
            meta.append((min(meta1[col][0],meta2[col][0]),max(meta1[col][1],meta2[col][1])))
    col += 1
    return meta

metadata = df.rdd.map(lambda row: metaMap(row,schema)).reduce(lambda x,y:metaReduce(x,y,schema))

def setToList(v):
    if isinstance(v,set):
        return list(v)
    return v

metadata = map(lambda x: setToList(x), metadata)
print metadata
```

```

lookup = {}
for i in range(0,len(schema)):
    lookup[schema[i][0]] = i

def row2LabeledPoint(dm,lookup,target,predictors,row):
    target_index = lookup[target]
    tval = dm[target_index].index(row[target_index])
    pvals = []
    for predictor in predictors:
        predictor_index = lookup[predictor]
        if isinstance(dm[predictor_index],list):
            pval = dm[predictor_index].index(row[predictor_index])
        else:
            pval = row[predictor_index]
        pvals.append(pval)
    return LabeledPoint(tval,DenseVector(pvals))

# count number of target classes
predictorClassCount = len(metadata[lookup[target]])

# define function to extract categorical predictor information from datamodel
def getCategoricalFeatureInfo(dm,lookup,predictors):
    info = {}
    for i in range(0,len(predictors)):
        predictor = predictors[i]
        predictor_index = lookup[predictor]
        if isinstance(dm[predictor_index],list):
            info[i] = len(dm[predictor_index])
    return info

# convert dataframe to an RDD containing LabeledPoint
lps = df.rdd.map(lambda row: row2LabeledPoint(metadata,lookup,target,predictors,row))

treeModel = DecisionTree.trainClassifier(
    lps,
    numClasses=predictorClassCount,
    categoricalFeaturesInfo=getCategoricalFeatureInfo(metadata, lookup, predictors),
    impurity='gini',
    maxDepth=5,
    maxBins=100)

_outputPath = ctxt.createTemporaryFolder()
treeModel.save(ctxt.getSparkContext(), _outputPath)
ctxt.setModelContentFromPath("TreeModel", _outputPath)
ctxt.setModelContentFromString("model.dm",json.dumps(metadata), mimeType="application/json") \
.setModelContentFromString("model.structure",treeModel.toDebugString())

"""

node.setPropertyValue("python_build_syntax", build_script)

```

R example

```

##### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_build_syntax", """modelerModel <- lm(modelerData$Na~modelerData$K,modelerData)
modelerDataModel
modelerModel
""")

```

Table 1. extensionmodelnode properties

extensionmodelnode Properties	Values	Property description
<code>syntax_type</code>	<code>R Python</code>	Specify which script runs – R or Python (R is the default).
<code>r_build_syntax</code>	<code>string</code>	The R scripting syntax for model building.
<code>r_score_syntax</code>	<code>string</code>	The R scripting syntax for model scoring.
<code>python_build_syntax</code>	<code>string</code>	The Python scripting syntax for model building.
<code>python_score_syntax</code>	<code>string</code>	The Python scripting syntax for model scoring.
<code>convert_flags</code>	<code>StringsAndDouble s LogicalValues</code>	Option to convert flag fields.
<code>convert_missing</code>	<code>flag</code>	Option to convert missing values to R NA value.
<code>convert_datetime</code>	<code>flag</code>	Option to convert variables with date or datetime formats to R date/time formats.
<code>convert_datetime_class</code>	<code>POSIXct POSIXlt</code>	Options to specify to what format variables with date or datetime formats are converted.
<code>output_html</code>	<code>flag</code>	Option to display graphs on a tab in the R model nugget.

extensionmodelnode Properties	Values	Property description
output_text	<i>flag</i>	Option to write R console text output to a tab in the R model nugget.

factornode properties

	The PCA/Factor node provides powerful data-reduction techniques to reduce the complexity of your data. Principal components analysis (PCA) finds linear combinations of the input fields that do the best job of capturing the variance in the entire set of fields, where the components are orthogonal (perpendicular) to each other. Factor analysis attempts to identify underlying factors that explain the pattern of correlations within a set of observed fields. For both approaches, the goal is to find a small number of derived fields that effectively summarizes the information in the original set of fields.
---	--

Example

```
node = stream.create("factor", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["BP", "Na", "K"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Factor_Age")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("method", "GLS")
# Expert options
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("complete_records", True)
node.setPropertyValue("matrix", "Covariance")
node.setPropertyValue("max_iterations", 30)
node.setPropertyValue("extract_factors", "ByFactors")
node.setPropertyValue("min_eigenvalue", 3.0)
node.setPropertyValue("max_factor", 7)
node.setPropertyValue("sort_values", True)
node.setPropertyValue("hide_values", True)
node.setPropertyValue("hide_below", 0.7)
# "Rotation" section
node.setPropertyValue("rotation", "DirectOblimin")
node.setPropertyValue("delta", 0.3)
node.setPropertyValue("kappa", 7.0)
```

Table 1. factornode properties

factornode Properties	Values	Property description
inputs	[field1 ... fieldN]	PCA/Factor models use a list of input fields, but no target. Weight and frequency fields are not used. See the topic Common modeling node properties for more information.
method	PC ULS GLS ML PAF Alpha Image	
mode	Simple Expert	
max_iterations	number	
complete_records	flag	
matrix	Correlation Covariance	
extract_factors	ByEigenvalues ByFactors	
min_eigenvalue	number	
max_factor	number	
rotation	None Varimax DirectOblimin Equamax Quartimax Promax	
delta	number	If you select DirectOblimin as your rotation data type, you can specify a value for delta . If you do not specify a value, the default value for delta is used.
kappa	number	If you select Promax as your rotation data type, you can specify a value for kappa . If you do not specify a value, the default value for kappa is used.
sort_values	flag	
hide_values	flag	
hide_below	number	

featureselectionnode properties



The Feature Selection node screens input fields for removal based on a set of criteria (such as the percentage of missing values); it then ranks the importance of remaining inputs relative to a specified target. For example, given a data set with hundreds of potential inputs, which are most likely to be useful in modeling patient outcomes?

Example

```
node = stream.create("featureselection", "My node")
node.setPropertyValue("screen_single_category", True)
node.setPropertyValue("max_single_category", 95)
node.setPropertyValue("screen_missing_values", True)
node.setPropertyValue("max_missing_values", 80)
node.setPropertyValue("criteria", "Likelihood")
node.setPropertyValue("unimportant_below", 0.8)
node.setPropertyValue("important_above", 0.9)
node.setPropertyValue("important_label", "Check Me Out!")
node.setPropertyValue("selection_mode", "TopN")
node.setPropertyValue("top_n", 15)
```

For a more detailed example that creates and applies a Feature Selection model, see [in](#).

Table 1. featureselectionnode properties

featureselectionnode Properties	Values	Property description
<code>target</code>	<code>field</code>	Feature Selection models rank predictors relative to the specified target. Weight and frequency fields are not used. See the topic Common modeling node properties for more information.
<code>screen_single_category</code>	<code>flag</code>	If <code>True</code> , screens fields that have too many records falling into the same category relative to the total number of records.
<code>max_single_category</code>	<code>number</code>	Specifies the threshold used when <code>screen_single_category</code> is <code>True</code> .
<code>screen_missing_values</code>	<code>flag</code>	If <code>True</code> , screens fields with too many missing values, expressed as a percentage of the total number of records.
<code>max_missing_values</code>	<code>number</code>	
<code>screen_num_categories</code>	<code>flag</code>	If <code>True</code> , screens fields with too many categories relative to the total number of records.
<code>max_num_categories</code>	<code>number</code>	
<code>screen_std_dev</code>	<code>flag</code>	If <code>True</code> , screens fields with a standard deviation of less than or equal to the specified minimum.
<code>min_std_dev</code>	<code>number</code>	
<code>screen_coeff_of_var</code>	<code>flag</code>	If <code>True</code> , screens fields with a coefficient of variance less than or equal to the specified minimum.
<code>min_coeff_of_var</code>	<code>number</code>	
<code>criteria</code>	<code>Pearson Likelihood</code> <code>Cramers V Lambda</code>	When ranking categorical predictors against a categorical target, specifies the measure on which the importance value is based.
<code>unimportant_below</code>	<code>number</code>	Specifies the threshold p values used to rank variables as important, marginal, or unimportant. Accepts values from 0.0 to 1.0.
<code>important_above</code>	<code>number</code>	Accepts values from 0.0 to 1.0.
<code>unimportant_label</code>	<code>string</code>	Specifies the label for the unimportant ranking.
<code>marginal_label</code>	<code>string</code>	
<code>important_label</code>	<code>string</code>	
<code>selection_mode</code>	<code>ImportanceLevel</code> <code>ImportanceValue</code> <code>TopN</code>	
<code>select_important</code>	<code>flag</code>	When <code>selection_mode</code> is set to <code>ImportanceLevel</code> , specifies whether to select important fields.
<code>select_marginal</code>	<code>flag</code>	When <code>selection_mode</code> is set to <code>ImportanceLevel</code> , specifies whether to select marginal fields.
<code>select_unimportant</code>	<code>flag</code>	When <code>selection_mode</code> is set to <code>ImportanceLevel</code> , specifies whether to select unimportant fields.
<code>importance_value</code>	<code>number</code>	When <code>selection_mode</code> is set to <code>ImportanceValue</code> , specifies the cutoff value to use. Accepts values from 0 to 100.

featureselecti onnode Properties	Values	Property description
top_n	<i>integer</i>	When selection_mode is set to TopN , specifies the cutoff value to use. Accepts values from 0 to 1000.

genlinnode properties

	The Generalized Linear model expands the general linear model so that the dependent variable is linearly related to the factors and covariates through a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers the functionality of a wide number of statistical models, including linear regression, logistic regression, loglinear models for count data, and interval-censored survival models.
---	--

Example

```
node = stream.create("genlin", "My node")
node.setPropertyValue("model_type", "MainAndAllTwoWayEffects")
node.setPropertyValue("offset_type", "Variable")
node.setPropertyValue("offset_field", "Claimant")
```

Table 1. genlinnode properties

genlinnode Properties	Values	Property description
target	<i>field</i>	Generalized Linear models require a single target field which must be a nominal or flag field, and one or more input fields. A weight field can also be specified. See the topic Common modeling node properties for more information.
use_weight	<i>flag</i>	
weight_field	<i>field</i>	Field type is only continuous.
target_repre sents_trials	<i>flag</i>	
trials_type	Variable FixedValue	
trials_field	<i>field</i>	Field type is continuous, flag, or ordinal.
trials_numbe r	<i>number</i>	Default value is 10.
model_type	MainEffects MainAndAllTwoWayEffects	
offset_type	Variable FixedValue	
offset_field	<i>field</i>	Field type is only continuous.
offset_value	<i>number</i>	Must be a real number.
base_categor y	LastFirst	
include_intercept	<i>flag</i>	
mode	Simple Expert	
distribution	BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIAL	IGAUSS : Inverse Gaussian. NEGBIN : Negative binomial.
negbin_para type	Specify Estimate	
negbin_param eter	<i>number</i>	Default value is 1. Must contain a non-negative real number.
tweedie_para meter	<i>number</i>	
link_funcatio n	IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPOWER PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT	CLOGLOG : Complementary log-log. LOGC : log complement. NEGBIN : Negative binomial. NLOGLOG : Negative log-log. CUMCAUCHIT : Cumulative cauchit. CUMCLOGLOG : Cumulative complementary log-log. CUMLOGIT : Cumulative logit. CUMNLOGLOG : Cumulative negative log-log. CUMPROBIT : Cumulative probit.
power	<i>number</i>	Value must be real, nonzero number.
method	Hybrid Fisher NewtonRaphson	
max_fisher_i terations	<i>number</i>	Default value is 1; only positive integers allowed.
scale_method	MaxLikelihoodEstimate Deviance PearsonChiSquare FixedValue	
scale_value	<i>number</i>	Default value is 1; must be greater than 0.
covariance_m atrix	ModelEstimator RobustEstimator	

genlinnode Properties	Values	Property description
max_iterations	number	Default value is 100; non-negative integers only.
max_step_halving	number	Default value is 5; positive integers only.
check_separation	flag	
start_iteration	number	Default value is 20; only positive integers allowed.
estimates_change	flag	
estimates_change_min	number	Default value is 1E-006; only positive numbers allowed.
estimates_change_type	Absolute Relative	
loglikelihood_change	flag	
loglikelihood_change_min	number	Only positive numbers allowed.
loglikelihood_change_type	Absolute Relative	
hessian_convergence	flag	
hessian_convergence_min	number	Only positive numbers allowed.
hessian_convergence_type	Absolute Relative	
case_summary	flag	
contrast_matrixes	flag	
descriptive_statistics	flag	
estimable_functions	flag	
model_info	flag	
iteration_history	flag	
goodness_of_fit	flag	
print_interval	number	Default value is 1; must be positive integer.
model_summary	flag	
lagrange_multiplier	flag	
parameter_estimates	flag	
include_exponential	flag	
covariance_estimates	flag	
correlation_estimates	flag	
analysis_type	TypeI TypeIII TypeIAndTypeIII	
statistics	Wald LR	
ctype	Wald Profile	
tolerancelevel	number	Default value is 0.0001.
confidence_interval	number	Default value is 95.
loglikelihood_function	Full Kernel	
singularity_tolerance	1E-007 1E-008 1E-009 1E-010 1E-011 1E-012	
value_order	Ascending Descending DataOrder	
calculate_variable_importance	flag	

genlinnode Properties	Values	Property description
calculate_ra w_propensiti es	flag	
calculate_ad justed_prope nsities	flag	
adjusted_pro pensity_part ition	Test Validation	

glmmnode properties

	A generalized linear mixed model (GLMM) extends the linear model so that the target can have a non-normal distribution, is linearly related to the factors and covariates via a specified link function, and so that the observations can be correlated. Generalized linear mixed models cover a wide variety of models, from simple linear regression to complex multilevel models for non-normal longitudinal data.
---	---

Table 1. glmmnode properties

glmmnode Properties	Values	Property description
residual_sub ject_spec	structured	The combination of values of the specified categorical fields that uniquely define subjects within the data set
repeated_me asures	structured	Fields used to identify repeated observations.
residual_gro up_spec	[field1 ... fieldN]	Fields that define independent sets of repeated effects covariance parameters.
residual_cov ariance_type	Diagonal AR1 ARMA11 COMPOUND_SYMME TRY IDENTITY TOEPLITZ UNSTRUCTURED VARIANCE_COMPONENTS	Specifies covariance structure for residuals.
custom_targe t	flag	Indicates whether to use target defined in upstream node (false) or custom target specified by target_field (true).
target_field	field	Field to use as target if custom_target is true .
use_trials	flag	Indicates whether additional field or value specifying number of trials is to be used when target response is a number of events occurring in a set of trials. Default is false .
use_field_or _value	FieldValue	Indicates whether field (default) or value is used to specify number of trials.
trials_field	field	Field to use to specify number of trials.
trials_value	integer	Value to use to specify number of trials. If specified, minimum value is 1.
use_custom_t arget_referen ce	flag	Indicates whether custom reference category is to be used for a categorical target. Default is false .
target_refer ence_value	string	Reference category to use if use_custom_target_reference is true .
dist_link_co mbination	Nominal Logit Gamma Log Binomial Logit Poisson Log Binomial Probit Negbin Log Binomial LogC Custom	Common models for distribution of values for target. Choose Custom to specify a distribution from the list provided by target_distribution .
target_distr ibution	Normal Binomial Multinomial Gamma Inverse NegativeBinomial Poisson	Distribution of values for target when dist_link_combination is Custom .
link_funcatio n_type	Identity LogC Log CLOGLOG Logit NLOGLOG PROBIT POWER CAUCHIT	Link function to relate target values to predictors. If target_distribution is Binomial you can use any of the listed link functions. If target_distribution is Multinomial you can use CLOGLOG , CAUCHIT , LOGIT , NLOGLOG , or PROBIT . If target_distribution is anything other than Binomial or Multinomial you can use IDENTITY , LOG , or POWER .
link_funcatio n_param	number	Link function parameter value to use. Only applicable if normal_link_function or link_function_type is POWER .

glmmnode Properties	Values	Property description
<code>use_predefined_inputs</code>	<code>flag</code>	Indicates whether fixed effect fields are to be those defined upstream as input fields (<code>true</code>) or those from <code>fixed_effects_list</code> (<code>false</code>). Default is <code>false</code> .
<code>fixed_effects_list</code>	<code>structured</code>	If <code>use_predefined_inputs</code> is <code>false</code> , specifies the input fields to use as fixed effect fields.
<code>use_intercept</code>	<code>flag</code>	If <code>true</code> (default), includes the intercept in the model.
<code>random_effecs_list</code>	<code>structured</code>	List of fields to specify as random effects.
<code>regression_weight_field</code>	<code>field</code>	Field to use as analysis weight field.
<code>use_offset</code>	<code>None offset_value offset_field</code>	Indicates how offset is specified. Value <code>None</code> means no offset is used.
<code>offset_value</code>	<code>number</code>	Value to use for offset if <code>use_offset</code> is set to <code>offset_value</code> .
<code>offset_field</code>	<code>field</code>	Field to use for offset value if <code>use_offset</code> is set to <code>offset_field</code> .
<code>target_categorical_order</code>	<code>Ascending Descending Data</code>	Sorting order for categorical targets. Value <code>Data</code> specifies using the sort order found in the data. Default is <code>Ascending</code> .
<code>inputs_categorical_order</code>	<code>Ascending Descending Data</code>	Sorting order for categorical predictors. Value <code>Data</code> specifies using the sort order found in the data. Default is <code>Ascending</code> .
<code>max_iterations</code>	<code>integer</code>	Maximum number of iterations the algorithm will perform. A non-negative integer; default is 100.
<code>confidence_level</code>	<code>integer</code>	Confidence level used to compute interval estimates of the model coefficients. A non-negative integer; maximum is 100, default is 95.
<code>degrees_of_freedom_method</code>	<code>Fixed Varied</code>	Specifies how degrees of freedom are computed for significance test.
<code>test_fixed_effects_coefficients</code>	<code>Model Robust</code>	Method for computing the parameter estimates covariance matrix.
<code>use_p_converge</code>	<code>flag</code>	Option for parameter convergence.
<code>p_converge</code>	<code>number</code>	Blank, or any positive value.
<code>p_converge_type</code>	<code>Absolute Relative</code>	
<code>use_l_converge</code>	<code>flag</code>	Option for log-likelihood convergence.
<code>l_converge</code>	<code>number</code>	Blank, or any positive value.
<code>l_converge_type</code>	<code>Absolute Relative</code>	
<code>use_h_converge</code>	<code>flag</code>	Option for Hessian convergence.
<code>h_converge</code>	<code>number</code>	Blank, or any positive value.
<code>h_converge_type</code>	<code>Absolute Relative</code>	
<code>max_fisher_steps</code>	<code>integer</code>	
<code>singularity_tolerance</code>	<code>number</code>	
<code>use_model_name</code>	<code>flag</code>	Indicates whether to specify a custom name for the model (<code>true</code>) or to use the system-generated name (<code>false</code>). Default is <code>false</code> .
<code>model_name</code>	<code>string</code>	If <code>use_model_name</code> is <code>true</code> , specifies the model name to use.
<code>confidence</code>	<code>onProbability onIncrease</code>	Basis for computing scoring confidence value: highest predicted probability, or difference between highest and second highest predicted probabilities.
<code>score_categorical_probabilities</code>	<code>flag</code>	If <code>true</code> , produces predicted probabilities for categorical targets. Default is <code>false</code> .
<code>max_categories</code>	<code>integer</code>	If <code>score_categorical_probabilities</code> is <code>true</code> , specifies maximum number of categories to save.
<code>score_propensity</code>	<code>flag</code>	If <code>true</code> , produces propensity scores for flag target fields that indicate likelihood of "true" outcome for field.
<code>emeans</code>	<code>structure</code>	For each categorical field from the fixed effects list, specifies whether to produce estimated marginal means.
<code>covariance_list</code>	<code>structure</code>	For each continuous field from the fixed effects list, specifies whether to use the mean or a custom value when computing estimated marginal means.

glmmnode Properties	Values	Property description
mean_scale	Original Transformed	Specifies whether to compute estimated marginal means based on the original scale of the target (default) or on the link function transformation.
comparison_adjustment_method	LSD SEQBONFERRONI SEQSIDAK	Adjustment method to use when performing hypothesis tests with multiple contrasts.

gle properties



A GLE extends the linear model so that the target can have a non-normal distribution, is linearly related to the factors and covariates via a specified link function, and so that the observations can be correlated. Generalized linear mixed models cover a wide variety of models, from simple linear regression to complex multilevel models for non-normal longitudinal data.

Table 1. gle properties

gle Properties	Values	Property description
custom_target	flag	Indicates whether to use target defined in upstream node (<code>false</code>) or custom target specified by <code>target_field</code> (<code>true</code>).
target_field	field	Field to use as target if <code>custom_target</code> is <code>true</code> .
use_trials	flag	Indicates whether additional field or value specifying number of trials is to be used when target response is a number of events occurring in a set of trials. Default is <code>false</code> .
use_trials_field_or_value	FieldValue	Indicates whether field (default) or value is used to specify number of trials.
trials_field	field	Field to use to specify number of trials.
trials_value	integer	Value to use to specify number of trials. If specified, minimum value is 1.
use_custom_target_reference	flag	Indicates whether custom reference category is to be used for a categorical target. Default is <code>false</code> .
target_reference_value	string	Reference category to use if <code>use_custom_target_reference</code> is <code>true</code> .
dist_link_combination	NormalIdentity GammaLog PoissonLog NegbinLog TweedieIdentity NominalLogit BinomialLogit BinomialProbit BinomialLogC CUSTOM	Common models for distribution of values for target. Choose <code>CUSTOM</code> to specify a distribution from the list provided by <code>target_distribution</code> .
target_distribution	Normal Binomial Multinomial Gamma INVERSE_GAUSS NEG_BINOMIAL Poisson TWEEDIE UNKNOWN	Distribution of values for target when <code>dist_link_combination</code> is <code>Custom</code> .
link_function_type	UNKNOWN IDENTITY LOG LOGIT PROBIT COMPL_LOG_LOG POWER LOG_COMPL NEG_LOG_LOG ODDS_POWER NEG_BINOMIAL GEN_LOGIT CUMUL_LOGIT CUMUL_PROBIT CUMUL_COMPL_LOG_LOG CUMUL_NEG_LOG_LOG CUMUL_CAUCHIT	Link function to relate target values to predictors. If <code>target_distribution</code> is <code>Binomial</code> you can use: UNKNOWN IDENTITY LOG LOGIT PROBIT COMPL_LOG_LOG POWER LOG_COMPL NEG_LOG_LOG ODDS_POWER If <code>target_distribution</code> is <code>NEG_BINOMIAL</code> you can use: <code>NEG_BINOMIAL</code> . If <code>target_distribution</code> is <code>UNKNOWN</code> , you can use: <code>GEN_LOGIT CUMUL_LOGIT CUMUL_PROBIT CUMUL_COMPL_LOG_LOG CUMUL_NEG_LOG_LOG CUMUL_CAUCHIT</code>
link_function_param	number	Tweedie parameter value to use. Only applicable if <code>normal_link_function</code> or <code>link_function_type</code> is <code>POWER</code> .
tweedie_param	number	Link function parameter value to use. Only applicable if <code>dist_link_combination</code> is set to <code>TweedieIdentity</code> , or <code>link_function_type</code> is <code>TWEEDIE</code> .
use_predefined_inputs	flag	Indicates whether model effect fields are to be those defined upstream as input fields (<code>true</code>) or those from <code>fixed_effects_list</code> (<code>false</code>).
model_effect_list	structured	If <code>use_predefined_inputs</code> is <code>false</code> , specifies the input fields to use as model effect fields.
use_intercept	flag	If <code>true</code> (default), includes the intercept in the model.
regression_weight_field	field	Field to use as analysis weight field.
use_offset	None Value Variable	Indicates how offset is specified. Value <code>None</code> means no offset is used.
offset_value	number	Value to use for offset if <code>use_offset</code> is set to <code>offset_value</code> .
offset_field	field	Field to use for offset value if <code>use_offset</code> is set to <code>offset_field</code> .

gle Properties	Values	Property description
target_categ ory_order	Ascending Descending	Sorting order for categorical targets. Default is Ascending .
inputs_categ ory_order	Ascending Descending	Sorting order for categorical predictors. Default is Ascending .
max_iterations	integer	Maximum number of iterations the algorithm will perform. A non-negative integer; default is 100.
confidence_level	number	Confidence level used to compute interval estimates of the model coefficients. A non-negative integer; maximum is 100, default is 95.
test_fixed_effects_coefficients	Model Robust	Method for computing the parameter estimates covariance matrix.
detect_outliers	flag	When true the algorithm finds influential outliers for all distributions except multinomial distribution.
conduct_trend_analysis	flag	When true the algorithm conducts trend analysis for the scatter plot.
estimation_method	FISHER_SCORING NEWTON_RAPHSON HYBRID	Specify the maximum likelihood estimation algorithm.
max_fisher_iterations	integer	If using the FISHER_SCORING estimation_method , the maximum number of iterations. Minimum 0, maximum 20.
scale_parameter_method	MLE FIXED DEVIANCE PEARSON_CHISQUARE	Specify the method to be used for the estimation of the scale parameter.
scale_value	number	Only available if scale_parameter_method is set to Fixed .
negative_binomial_method	MLE FIXED	Specify the method to be for the estimation of the negative binomial ancillary parameter.
negative_binomial_value	number	Only available if negative_binomial_method is set to Fixed .
non_neg_least_squares	flag	Whether to perform non-negative least squares. Default is false .
use_p_converge	flag	Option for parameter convergence.
p_converge	number	Blank, or any positive value.
p_converge_type	flag	True = Absolute, False = Relative
use_l_converge	flag	Option for log-likelihood convergence.
l_converge	number	Blank, or any positive value.
l_converge_type	flag	True = Absolute, False = Relative
use_h_converge	flag	Option for Hessian convergence.
h_converge	number	Blank, or any positive value.
h_converge_type	flag	True = Absolute, False = Relative
max_iterations	integer	Maximum number of iterations the algorithm will perform. A non-negative integer; default is 100.
sing_tolerance	integer	
use_model_selection	flag	Enables the parameter threshold and model selection method controls..
method	LASSO ELASTIC_NET FORWARD_STEPWISE RIDGE	Determines the model selection method, or if using Ridge the regularization method, used.
detect_two_way_interactions	flag	When True the model will automatically detect two-way interactions between input fields. This control should only be enabled if the model is main effects only (that is, where the user has not created any higher order effects) and if the method selected is Forward Stepwise, Lasso, or Elastic Net.
automatic_penalty_params	flag	Only available if model selection method is Lasso or Elastic Net. Use this function to enter penalty parameters associated with either the Lasso or Elastic Net variable selection methods. If True , default values are used. If False , the penalty parameters are enabled custom values can be entered.
lasso_penalty_param	number	Only available if model selection method is Lasso or Elastic Net and automatic_penalty_params is False . Specify the penalty parameter value for Lasso.

gle Properties	Values	Property description
<code>elastic_net_penalty_params1</code>	<code>number</code>	Only available if model selection <code>method</code> is Lasso or Elastic Net and <code>automatic_penalty_params</code> is <code>False</code> . Specify the penalty parameter value for Elastic Net parameter 1.
<code>elastic_net_penalty_params2</code>	<code>number</code>	Only available if model selection <code>method</code> is Lasso or Elastic Net and <code>automatic_penalty_params</code> is <code>False</code> . Specify the penalty parameter value for Elastic Net parameter 2.
<code>probability_entry</code>	<code>number</code>	Only available if the <code>method</code> selected is Forward Stepwise. Specify the significance level of the f statistic criterion for effect inclusion.
<code>probability_removal</code>	<code>number</code>	Only available if the <code>method</code> selected is Forward Stepwise. Specify the significance level of the f statistic criterion for effect removal.
<code>use_max_effects</code>	<code>flag</code>	Only available if the <code>method</code> selected is Forward Stepwise. Enables the <code>max_effects</code> control. When <code>False</code> the default number of effects included should equal the total number of effects supplied to the model, minus the intercept.
<code>max_effects</code>	<code>integer</code>	Specify the maximum number of effects when using the forward stepwise building method.
<code>use_max_steps</code>	<code>flag</code>	Enables the <code>max_steps</code> control. When <code>False</code> the default number of steps should equal three times the number of effects supplied to the model, excluding the intercept.
<code>max_steps</code>	<code>integer</code>	Specify the maximum number of steps to be taken when using the Forward Stepwise building <code>method</code> .
<code>use_model_name</code>	<code>flag</code>	Indicates whether to specify a custom name for the model (<code>true</code>) or to use the system-generated name (<code>false</code>). Default is <code>false</code> .
<code>model_name</code>	<code>string</code>	If <code>use_model_name</code> is <code>true</code> , specifies the model name to use.
<code>usePI</code>	<code>flag</code>	If <code>true</code> , predictor importance is calculated..

kmeansnode properties



The K-Means node clusters the data set into distinct groups (or clusters). The method defines a fixed number of clusters, iteratively assigns records to clusters, and adjusts the cluster centers until further refinement can no longer improve the model. Instead of trying to predict an outcome, k-means uses a process known as unsupervised learning to uncover patterns in the set of input fields.

Example

```
node = stream.create("kmeans", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["Cholesterol", "BP", "Drug", "Na", "K", "Age"])
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Kmeans_allinputs")
node.setPropertyValue("num_clusters", 9)
node.setPropertyValue("gen_distance", True)
node.setPropertyValue("cluster_label", "Number")
node.setPropertyValue("label_prefix", "Kmeans_")
node.setPropertyValue("optimize", "Speed")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("stop_on", "Custom")
node.setPropertyValue("max_iterations", 10)
node.setPropertyValue("tolerance", 3.0)
node.setPropertyValue("encoding_value", 0.3)
```

Table 1. kmeansnode properties

kmeansnode Properties	Values	Property description
<code>inputs</code>	<code>[field1 ... fieldN]</code>	K-means models perform cluster analysis on a set of input fields but do not use a target field. Weight and frequency fields are not used. See the topic Common modeling node properties for more information.
<code>num_clusters</code>	<code>number</code>	
<code>gen_distance</code>	<code>flag</code>	
<code>cluster_label</code>	<code>String</code> <code>Number</code>	
<code>label_prefix</code>	<code>string</code>	
<code>mode</code>	<code>Simple</code> <code>Expert</code>	

kmeansnode Properties	Values	Property description
stop_on	Default Custom	
max_iterations	number	
tolerance	number	
encoding_value	number	
optimize	Speed Memory	Use to specify whether model building should be optimized for speed or for memory.

kmeansasnode properties



K-Means is one of the most commonly used clustering algorithms. It clusters data points into a predefined number of clusters. The K-Means-AS node in SPSS® Modeler is implemented in Spark. For details about K-Means algorithms, see <https://spark.apache.org/docs/2.2.0/ml-clustering.html>. Note that the K-Means-AS node performs one-hot encoding automatically for categorical variables.

Table 1. kmeansasnode properties

kmeansasnode Properties	Values	Property description
roleUse	string	Specify predefined to use predefined roles, or custom to use custom field assignments. Default is predefined .
autoModel	Boolean	Specify true to use the default name (\$S-prediction) for the new generated scoring field, or false to use a custom name. Default is true .
features	field	List of the field names for input when the roleUse property is set to custom .
name	string	The name of the new generated scoring field when the autoModel property is set to false .
clustersNum	integer	The number of clusters to create. Default is 5.
initMode	string	The initialization algorithm. Possible values are k-means or random . Default is k-means .
initSteps	integer	The number of initialization steps when initMode is set to k-means . Default is 2.
advancedSettings	Boolean	Specify true to make the following four properties available. Default is false .
maxIteration	integer	Maximum number of iterations for clustering. Default is 20.
tolerance	string	The tolerance to stop the iterations. Possible settings are 1.0E-1 , 1.0E-2 , ..., 1.0E-6 . Default is 1.0E-4 .
setSeed	Boolean	Specify true to use a custom random seed. Default is false .
randomSeed	integer	The custom random seed when the setSeed property is true .

knnnode properties



The *k*-Nearest Neighbor (KNN) node associates a new case with the category or value of the *k* objects nearest to it in the predictor space, where *k* is an integer. Similar cases are near each other and dissimilar cases are distant from each other.

Example

```
node = stream.create("knn", "My node")
# Objectives tab
node.setPropertyValue("objective", "Custom")
# Settings tab - Neighbors panel
node.setPropertyValue("automatic_k_selection", False)
node.setPropertyValue("fixed_k", 2)
node.setPropertyValue("weight_by_importance", True)
# Settings tab - Analyze panel
node.setPropertyValue("save_distances", True)
```

Table 1. knnnnode properties

knnnode Properties	Values	Property description
analysis	PredictTarget IdentifyNeighbors	
objective	Balance Speed Accuracy Custom	
normalize_ranges	flag	
use_case_labels	flag	Check box to enable next option.
case_labels_field	field	

knnnode Properties	Values	Property description
identify_focal_cases	flag	Check box to enable next option.
focal_cases_field	field	
automatic_k_selection	flag	
fixed_k	integer	Enabled only if automatic_k_selection is False.
minimum_k	integer	Enabled only if automatic_k_selection is True.
maximum_k	integer	
distance_computation	Euclidean CityBlock	
weight_by_importance	flag	
range_predictions	Mean Median	
perform_feature_selection	flag	
forced_entry_inputs	[field1 ... fieldN]	
stop_on_error_ratio	flag	
number_to_select	integer	
minimum_change	number	
validation_fold_assign_by_field	flag	
number_of_folds	integer	Enabled only if validation_fold_assign_by_field is False
set_random_seed	flag	
random_seed	number	
folds_field	field	Enabled only if validation_fold_assign_by_field is True
all_probabilities	flag	
save_distances	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

kohonennode properties



The Kohonen node generates a type of neural network that can be used to cluster the data set into distinct groups. When the network is fully trained, records that are similar should be close together on the output map, while records that are different will be far apart. You can look at the number of observations captured by each unit in the model nugget to identify the strong units. This may give you a sense of the appropriate number of clusters.

Example

```
node = stream.create("kohonen", "My node")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Symbolic Cluster")
node.setPropertyValue("stop_on", "Time")
node.setPropertyValue("time", 1)
node.setPropertyValue("set_random_seed", True)
node.setPropertyValue("random_seed", 12345)
node.setPropertyValue("optimize", "Speed")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("width", 3)
node.setPropertyValue("length", 3)
node.setPropertyValue("decay_style", "Exponential")
node.setPropertyValue("phasel_neighborhood", 3)
node.setPropertyValue("phasel_eta", 0.5)
node.setPropertyValue("phasel_cycles", 10)
node.setPropertyValue("phase2_neighborhood", 1)
node.setPropertyValue("phase2_eta", 0.2)
node.setPropertyValue("phase2_cycles", 75)
```

Table 1. kohonennode properties

kohonennode Properties	Values	Property description
inputs	[field1 ... fieldN]	Kohonen models use a list of input fields, but no target. Frequency and weight fields are not used. See the topic Common modeling node properties for more information.
continue	flag	

kohonennode Properties	Values	Property description
show_feedback	flag	
stop_on_time	Default Time	
optimize	Speed Memory	Use to specify whether model building should be optimized for speed or for memory.
cluster_label	flag	
mode	Simple Expert	
width	number	
length	number	
decay_style	Linear Exponential	
phase1_neighborhood	number	
phase1_eta	number	
phase1_cycles	number	
phase2_neighborhood	number	
phase2_eta	number	
phase2_cycles	number	

linarnode properties



Linear regression models predict a continuous target based on linear relationships between the target and one or more predictors.

Example

```
node = stream.create("linear", "My node")
# Build Options tab - Objectives panel
node.setPropertyValue("objective", "Standard")
# Build Options tab - Model Selection panel
node.setPropertyValue("model_selection", "BestSubsets")
node.setPropertyValue("criteria_best_subsets", "ASE")
# Build Options tab - Ensembles panel
node.setPropertyValue("combining_rule_categorical", "HighestMeanProbability")
```

Table 1. linarnode properties

linarnode Properties	Values	Property description
target	field	Specifies a single target field.
inputs	[field1 ... fieldN]	Predictor fields used by the model.
continue_training_existing_model	flag	
objective	Standard Bagging Boosting psm	psm is used for very large datasets, and requires a Server connection.
use_auto_data_preparation	flag	
confidence_level	number	
model_selection	Forward Stepwise BestSubsets None	
criteria_forward_stepwise	AICC Fstatistics AdjustedRSquare ASE	
probability_entry	number	
probability_removal	number	
use_max_effects	flag	
max_effects	number	
use_max_steps	flag	
max_steps	number	
criteria_best_subsets	AICC AdjustedRSquare ASE	
combining_rule_continuous	Mean Median	
component_models_n	number	
use_random_seed	flag	

linearnode Properties	Values	Property description
random_seed	number	
use_custom_model_name	flag	
custom_model_name	string	
use_custom_name	flag	
custom_name	string	
tooltip	string	
keywords	string	
annotation	string	

linearasnode properties



Linear regression models predict a continuous target based on linear relationships between the target and one or more predictors.

Table 1. linearasnode properties

linearasnode Properties	Values	Property description
target	field	Specifies a single target field.
inputs	[field1 ... fieldN]	Predictor fields used by the model.
weight_field	field	Analysis field used by the model.
custom_fields	flag	The default value is TRUE .
intercept	flag	The default value is TRUE .
detect_2way_interaction	flag	Whether or not to consider two way interaction. The default value is TRUE .
cin	number	The interval of confidence used to compute estimates of the model coefficients. Specify a value greater than 0 and less than 100. The default value is 95 .
factor_order	ascending descending	The sort order for categorical predictors. The default value is ascending .
var_select_method	ForwardStepwise BestSubsets none	The model selection method to use. The default value is ForwardStepwise .
criteria_for_forward_stepwise	AICC Fstatistics AdjustedRSquare ASE	The statistic used to determine whether an effect should be added to or removed from the model. The default value is AdjustedRSquare .
pin	number	The effect that has the smallest p-value less than this specified pin threshold is added to the model. The default value is 0 . 05 .
pout	number	Any effects in the model with a p-value greater than this specified pout threshold are removed. The default value is 0 . 10 .
use_custom_max_effects	flag	Whether to use max number of effects in the final model. The default value is FALSE .
max_effects	number	Maximum number of effects to use in the final model. The default value is 1 .
use_custom_max_steps	flag	Whether to use the maximum number of steps. The default value is FALSE .
max_steps	number	The maximum number of steps before the stepwise algorithm stops. The default value is 1 .
criteria_for_best_subsets	AICC AdjustedRSquare ASE	The mode of criteria to use. The default value is AdjustedRSquare .

logregnode properties



Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric range.

Multinomial Example

```
node = stream.create("logreg", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["BP", "Cholesterol", "Age"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Log_reg Drug")
node.setPropertyValue("use_partitioned_data", True)
```

```

node.setPropertyValue("method", "Stepwise")
node.setPropertyValue("logistic_procedure", "Multinomial")
node.setPropertyValue("multinomial_base_category", "BP")
node.setPropertyValue("model_type", "FullFactorial")
node.setPropertyValue("custom_terms", [[["BP", "Sex"], ["Age"], ["Na", "K"]]])
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("scale", "Pearson")
node.setPropertyValue("scale_value", 3.0)
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("tolerance", "1.0E-7")
# "Convergence..." section
node.setPropertyValue("max_iterations", 50)
node.setPropertyValue("max_steps", 3)
node.setPropertyValue("l_converge", "1.0E-3")
node.setPropertyValue("p_converge", "1.0E-7")
node.setPropertyValue("delta", 0.03)
# "Output..." section
node.setPropertyValue("summary", True)
node.setPropertyValue("likelihood_ratio", True)
node.setPropertyValue("asymptotic_correlation", True)
node.setPropertyValue("goodness_fit", True)
node.setPropertyValue("iteration_history", True)
node.setPropertyValue("history_steps", 3)
node.setPropertyValue("parameters", True)
node.setPropertyValue("confidence_interval", 90)
node.setPropertyValue("asymptotic_covariance", True)
node.setPropertyValue("classification_table", True)
# "Stepping" options
node.setPropertyValue("min_terms", 7)
node.setPropertyValue("use_max_terms", True)
node.setPropertyValue("max_terms", 10)
node.setPropertyValue("probability_entry", 3)
node.setPropertyValue("probability_removal", 5)
node.setPropertyValue("requirements", "Containment")

```

Binomial Example

```

node = stream.create("logreg", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Cholesterol")
node.setPropertyValue("inputs", ["BP", "Drug", "Age"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Log_reg Cholesterol")
node.setPropertyValue("multinomial_base_category", "BP")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("binomial_method", "Forwards")
node.setPropertyValue("logistic_procedure", "Binomial")
node.setPropertyValue("binomial_categorical_input", "Sex")
node.setKeyedPropertyValue("binomial_input_contrast", "Sex", "Simple")
node.setKeyedPropertyValue("binomial_input_category", "Sex", "Last")
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("scale", "Pearson")
node.setPropertyValue("scale_value", 3.0)
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("tolerance", "1.0E-7")
# "Convergence..." section
node.setPropertyValue("max_iterations", 50)
node.setPropertyValue("l_converge", "1.0E-3")
node.setPropertyValue("p_converge", "1.0E-7")
# "Output..." section
node.setPropertyValue("binomial_output_display", "at_each_step")
node.setPropertyValue("binomial_goodness_of_fit", True)
node.setPropertyValue("binomial_iteration_history", True)
node.setPropertyValue("binomial_parameters", True)
node.setPropertyValue("binomial_ci_enable", True)
node.setPropertyValue("binomial_ci", 85)
# "Stepping" options
node.setPropertyValue("binomial_removal_criterion", "LR")
node.setPropertyValue("binomial_probability_removal", 0.2)

```

Table 1. logregnode properties

logregnode Properties	Values	Property description
-----------------------	--------	----------------------

logregnode Properties	Values	Property description
target	<i>field</i>	Logistic regression models require a single target field and one or more input fields. Frequency and weight fields are not used. See the topic Common modeling node properties for more information.
logistic_procedure	Binomial Multinomial	
include_constant	<i>flag</i>	
mode	Simple Expert	
method	Enter Stepwise Forwards Backwards BackwardsStepwise	
binomial_method	Enter Forwards Backwards	
model_type	MainEffects FullFactorial Custom	When FullFactorial is specified as the model type, stepping methods will not be run, even if specified. Instead, Enter will be the method used. If the model type is set to Custom but no custom fields are specified, a main-effects model will be built.
custom_terms	<i>[[BP Sex][BP][Age]]</i>	
multinomial_base_category	<i>string</i>	Specifies how the reference category is determined.
binomial_categorical_input	<i>string</i>	
binomial_input_contrast	Indicator Simple Difference Helmert Repeated Polynomial Deviation	Keyed property for categorical input that specifies how the contrast is determined. <i>See the example for usage.</i>
binomial_input_category	First Last	Keyed property for categorical input that specifies how the reference category is determined. <i>See the example for usage.</i>
scale	None UserDefined Pearson Deviance	
scale_value	<i>number</i>	
all_probabilities	<i>flag</i>	
tolerance	1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10	
min_terms	<i>number</i>	
use_max_terms	<i>flag</i>	
max_terms	<i>number</i>	
entry_criterion	Score LR	
removal_criterion	LR Wald	
probability_entry	<i>number</i>	
probability_removal	<i>number</i>	
binomial_probability_entry	<i>number</i>	
binomial_probability_removal	<i>number</i>	
requirements	HierarchyDiscrete HierarchyAll Containment None	
max_iterations	<i>number</i>	
max_steps	<i>number</i>	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
delta	<i>number</i>	
iteration_history	<i>flag</i>	

logregnode Properties	Values	Property description
history_steps	<i>number</i>	
summary	<i>flag</i>	
likelihood_ratio	<i>flag</i>	
asymptotic_correlation	<i>flag</i>	
goodness_fit	<i>flag</i>	
parameters	<i>flag</i>	
confidence_interval	<i>number</i>	
asymptotic_covariance	<i>flag</i>	
classification_table	<i>flag</i>	
stepwise_summary	<i>flag</i>	
info_criteria	<i>flag</i>	
monotonicity_measures	<i>flag</i>	
binomial_output_display	<i>at_each_step</i> <i>at_last_step</i>	
binomial_goodness_of_fit	<i>flag</i>	
binomial_parameters	<i>flag</i>	
binomial_iteration_history	<i>flag</i>	
binomial_classification_plots	<i>flag</i>	
binomial_ci_enable	<i>flag</i>	
binomial_ci	<i>number</i>	
binomial_residual	<i>outliers all</i>	
binomial_residual_enable	<i>flag</i>	
binomial_outlier_threshold	<i>number</i>	
binomial_classification_cutoff	<i>number</i>	
binomial_removeval_criterion	<i>LR Wald Conditional</i>	
calculate_variable_importance	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	

lsvmnode properties



The Linear Support Vector Machine (LSVM) node enables you to classify data into one of two groups without overfitting. LSVM is linear and works well with wide data sets, such as those with a very large number of records.

Table 1. lsvmnode properties

lsvmnode Properties	Values	Property description
intercept	<i>flag</i>	Includes the intercept in the model. Default value is True .
target_order	<i>Ascending</i> <i>Descending</i>	Specifies the sorting order for the categorical target. Ignored for continuous targets. Default is Ascending .

lsvmnode Properties	Values	Property description
precision	<i>number</i>	Used only if measurement level of target field is Continuous . Specifies the parameter related to the sensitiveness of the loss for regression. Minimum is 0 and there is no maximum. Default value is 0 . 1 .
exclude_missing_values	<i>flag</i>	When True , a record is excluded if any single value is missing. The default value is False .
penalty_function	L1 L2	Specifies the type of penalty function used. The default value is L2 .
lambda	<i>number</i>	Penalty (regularization) parameter.
calculate_variable_importance	<i>flag</i>	For models that produce an appropriate measure of importance, this option displays a chart that indicates the relative importance of each predictor in estimating the model. Note that variable importance may take longer to calculate for some models, particularly when working with large datasets, and is off by default for some models as a result. Variable importance is not available for decision list models.

neuralnetnode properties

Important: A newer version of the Neural Net modeling node, with enhanced features, is available in this release and is described in the next section (*neuralnetwork*). Although you can still build and score a model with the previous version, we recommend updating your scripts to use the new version. Details of the previous version are retained here for reference.

Example

```
node = stream.create("neuralnet", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("targets", ["Drug"])
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
# "Model" tab
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("method", "Dynamic")
node.setPropertyValue("train_pct", 30)
node.setPropertyValue("set_random_seed", True)
node.setPropertyValue("random_seed", 12345)
node.setPropertyValue("stop_on", "Time")
node.setPropertyValue("accuracy", 95)
node.setPropertyValue("cycles", 200)
node.setPropertyValue("time", 3)
node.setPropertyValue("optimize", "Speed")
# "Multiple Method Expert Options" section
node.setPropertyValue("m_topologies", "5 30 5; 2 20 3, 1 10 1")
node.setPropertyValue("m_non_pyramids", False)
node.setPropertyValue("m_persistence", 100)
```

Table 1. neuralnetnode properties

neuralnetnode Properties	Values	Property description
targets	<i>[field1 ... fieldN]</i>	The Neural Net node expects one or more target fields and one or more input fields. Frequency and weight fields are ignored. See the topic Common modeling node properties for more information.
method	Quick Dynamic Multiple Prune ExhaustivePrune RBFN	
prevent_overtrain	<i>flag</i>	
train_pct	<i>number</i>	
set_random_seed	<i>flag</i>	
random_seed	<i>number</i>	
mode	Simple Expert	
stop_on	Default Accuracy Cycles Time	Stopping mode.
accuracy	<i>number</i>	Stopping accuracy.
cycles	<i>number</i>	Cycles to train.
time	<i>number</i>	Time to train (minutes).
continue	<i>flag</i>	
show_feedback	<i>flag</i>	
binary_encode	<i>flag</i>	

neuralnetnode Properties	Values	Property description
<code>use_last_mode_1</code>	<code>flag</code>	
<code>gen_logfile</code>	<code>flag</code>	
<code>logfile_name</code>	<code>string</code>	
<code>alpha</code>	<code>number</code>	
<code>initial_eta</code>	<code>number</code>	
<code>high_eta</code>	<code>number</code>	
<code>low_eta</code>	<code>number</code>	
<code>eta_decay_cycles</code>	<code>number</code>	
<code>hid_layers</code>	<code>One Two Three</code>	
<code>hl_units_one</code>	<code>number</code>	
<code>hl_units_two</code>	<code>number</code>	
<code>hl_units_three</code>	<code>number</code>	
<code>persistence</code>	<code>number</code>	
<code>m_topologies</code>	<code>string</code>	
<code>m_non_pyramids</code>	<code>flag</code>	
<code>m_persistence</code>	<code>number</code>	
<code>p_hid_layers</code>	<code>One Two Three</code>	
<code>p_hl_units_one</code>	<code>number</code>	
<code>p_hl_units_two</code>	<code>number</code>	
<code>p_hl_units_three</code>	<code>number</code>	
<code>p_persistence</code>	<code>number</code>	
<code>p_hid_rate</code>	<code>number</code>	
<code>p_hid_pers</code>	<code>number</code>	
<code>p_inp_rate</code>	<code>number</code>	
<code>p_inp_pers</code>	<code>number</code>	
<code>p_overall_pers</code>	<code>number</code>	
<code>r_persistence</code>	<code>number</code>	
<code>r_num_clusters</code>	<code>number</code>	
<code>r_eta_auto</code>	<code>flag</code>	
<code>r_alpha</code>	<code>number</code>	
<code>r_eta</code>	<code>number</code>	
<code>optimize</code>	<code>Speed Memory</code>	Use to specify whether model building should be optimized for speed or for memory.
<code>calculate_variable_importance</code>	<code>flag</code>	Note: The <code>sensitivity_analysis</code> property used in previous releases is deprecated in favor of this property. The old property is still supported, but <code>calculate_variable_importance</code> is recommended.
<code>calculate_raw_propensities</code>	<code>flag</code>	
<code>calculate_adjusted_propensities</code>	<code>flag</code>	
<code>adjusted propensity_partition</code>	<code>Test Validation</code>	

neuralnetworknode properties

	The Neural Net node uses a simplified model of the way the human brain processes information. It works by simulating a large number of interconnected simple processing units that resemble abstract versions of neurons. Neural networks are powerful general function estimators and require minimal statistical or mathematical knowledge to train or apply.
---	---

Example

```
node = stream.create("neuralnetwork", "My node")
# Build Options tab - Objectives panel
```

```

node.setPropertyValue("objective", "Standard")
# Build Options tab - Ensembles panel
node.setPropertyValue("combining_rule_categorical", "HighestMeanProbability")

```

Table 1. neuralnetworknode properties

neuralnetworknode Properties	Values	Property description
targets	[field1 ... fieldN]	Specifies target fields.
inputs	[field1 ... fieldN]	Predictor fields used by the model.
splits	[field1 ... fieldN]	Specifies the field or fields to use for split modeling.
use_partition	flag	If a partition field is defined, this option ensures that only data from the training partition is used to build the model.
continue	flag	Continue training existing model.
objective	Standard Bagging Boosting psm	psm is used for very large datasets, and requires a Server connection.
method	MultilayerPerceptron RadialBasisFunction	
use_custom_layers	flag	
first_layer_units	number	
second_layer_units	number	
use_max_time	flag	
max_time	number	
use_max_cycles	flag	
max_cycles	number	
use_min_accuracy	flag	
min_accuracy	number	
combining_rule_categorical	Voting HighestProbability HighestMeanProbability	
combining_rule_continuous	Mean Median	
component_models_n	number	
overfit_prevention_pct	number	
use_random_seed	flag	
random_seed	number	
missing_values	listwiseDeletion missingValueImputation	
use_model_name	boolean	
model_name	string	
confidence	onProbability onIncrease	
score_category_probabilities	flag	
max_categories	number	
score_propensity	flag	
use_custom_name	flag	
custom_name	string	
tooltip	string	
keywords	string	
annotation	string	

questnode properties



The QUEST node provides a binary classification method for building decision trees, designed to reduce the processing time required for large C&R Tree analyses while also reducing the tendency found in classification tree methods to favor inputs that allow more splits. Input fields can be numeric ranges (continuous), but the target field must be categorical. All splits are binary.

Example

```

node = stream.create("quest", "My node")
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
node.setPropertyValue("model_output_type", "InteractiveBuilder")

```

```

node.setPropertyValue("use_tree_directives", True)
node.setPropertyValue("max_surrogates", 5)
node.setPropertyValue("split_alpha", 0.03)
node.setPropertyValue("use_percentage", False)
node.setPropertyValue("min_parent_records_abs", 40)
node.setPropertyValue("min_child_records_abs", 30)
node.setPropertyValue("prune_tree", True)
node.setPropertyValue("use_std_err", True)
node.setPropertyValue("std_err_multiplier", 3)

```

Table 1. questnode properties

questnode Properties	Values	Property description
target	<i>field</i>	QUEST models require a single target and one or more input fields. A frequency field can also be specified. See the topic Common modeling node properties for more information.
continue_training_existing_model	<i>flag</i>	
objective	Standard Boosting Bagging psm	psm is used for very large datasets, and requires a Server connection.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>flag</i>	
tree_directives	<i>string</i>	
use_max_depth	Default Custom	
max_depth	<i>integer</i>	Maximum tree depth, from 0 to 1000. Used only if use_max_depth = Custom .
prune_tree	<i>flag</i>	Prune tree to avoid overfitting.
use_std_err	<i>flag</i>	Use maximum difference in risk (in Standard Errors).
std_err_multiplier	<i>number</i>	Maximum difference.
max_surrogates	<i>number</i>	Maximum surrogates.
use_percentage	<i>flag</i>	
min_parent_records_pc	<i>number</i>	
min_child_records_pc	<i>number</i>	
min_parent_records_abs	<i>number</i>	
min_child_records_abs	<i>number</i>	
use_costs	<i>flag</i>	
costs	structured	Structured property.
priors	Data Equal Custom	
custom_priors	structured	Structured property.
adjust_priors	<i>flag</i>	
trails	<i>number</i>	Number of component models for boosting or bagging.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Default combining rule for categorical targets.
range_ensemble_method	Mean Median	Default combining rule for continuous targets.
large_boost	<i>flag</i>	Apply boosting to very large data sets.
split_alpha	<i>number</i>	Significance level for splitting.
train_pct	<i>number</i>	Overfit prevention set.
set_random_seed	<i>flag</i>	Replicate results option.
seed	<i>number</i>	
calculate_variable_importance	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_proportionality_partition	Test Validation	

randomtrees properties



The Random Trees node is similar to the existing C&RT node; however, the Random Trees node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS® Modeler version 17. The Random Trees tree node generates a decision tree that you use to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered *pure* if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).

Table 1. randomtrees properties

Properties	Values	Property description
target	<i>field</i>	In the Random Trees node, models require a single target and one or more input fields. A frequency field can also be specified. See the topic Common modeling node properties for more information.
number_of_models	<i>integer</i>	Determines the number of models to build as part of the ensemble modeling.
use_number_of_predictors	<i>flag</i>	Determines whether number_of_predictors is used.
number_of_predictors	<i>integer</i>	Specifies the number of predictors to be used when building split models.
use_stop_rule_for_accuracy	<i>flag</i>	Determines whether model building stops when accuracy cannot be improved.
sample_size	<i>number</i>	Reduce this value to improve performance when processing very large datasets.
handle_imbalanced_data	<i>flag</i>	If the target of the model is a particular flag outcome, and the ratio of the desired outcome to a non-desired outcome is very small, then the data is imbalanced and the bootstrap sampling that is conducted by the model may affect the model's accuracy. Enable imbalanced data handling so that the model will capture a larger proportion of the desired outcome and generate a stronger model.
use_weighted_sampling	<i>flag</i>	When <i>False</i> , variables for each node are randomly selected with the same probability. When <i>True</i> , variables are weighted and selected accordingly.
max_node_number	<i>integer</i>	Maximum number of nodes allowed in individual trees. If the number would be exceeded on the next split, tree growth halts.
max_depth	<i>integer</i>	Maximum tree depth before growth halts.
min_child_node_size	<i>integer</i>	Determines the minimum number of records allowed in a child node after the parent node is split. If a child node would contain fewer records than specified here the parent node will not be split
use_costs	<i>flag</i>	
costs	<i>structured</i>	Structured property. The format is a list of 3 values: the actual value, the predicted value, and the cost if that prediction is wrong. For example: tree.setPropertyValue("costs", [{"drugA": "drugB", 3.0}, {"drugX": "drugY", 4.0}])
default_cost_increase	<i>none linear square custom</i>	Note: only enabled for ordinal targets. Set default values in the costs matrix.
max_pct_missing	<i>integer</i>	If the percentage of missing values in any input is greater than the value specified here, the input is excluded. Minimum 0, maximum 100.
exclude_single_cat_pct	<i>integer</i>	If one category value represents a higher percentage of the records than specified here, the entire field is excluded from model building. Minimum 1, maximum 99.
max_category_number	<i>integer</i>	If the number of categories in a field exceeds this value, the field is excluded from model building. Minimum 2.
min_field_variation	<i>number</i>	If the coefficient of variation of a continuous field is smaller than this value, the field is excluded from model building.
num_bins	<i>integer</i>	Only used if the data is made up of continuous inputs. Set the number of equal frequency bins to be used for the inputs; options are: 2, 4, 5, 10, 20, 25, 50, or 100.
topN	<i>integer</i>	Specifies the number of rules to report. Default value is 50, with a minimum of 1 and a maximum of 1000.

regressionnode properties



Linear regression is a common statistical technique for summarizing data and making predictions by fitting a straight line or surface that minimizes the discrepancies between predicted and actual output values.

Note: The Regression node is due to be replaced by the Linear node in a future release. We recommend using Linear models for linear regression from now on.

Example

```

node = stream.create("regression", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Age")
node.setPropertyValue("inputs", ["Na", "K"])
node.setPropertyValue("partition", "Test")
node.setPropertyValue("use_weight", True)
node.setPropertyValue("weight_field", "Drug")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Regression Age")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("method", "Stepwise")
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("complete_records", False)
node.setPropertyValue("tolerance", "1.0E-3")
# "Stepping..." section
node.setPropertyValue("stepping_method", "Probability")
node.setPropertyValue("probability_entry", 0.77)
node.setPropertyValue("probability_removal", 0.88)
node.setPropertyValue("F_value_entry", 7.0)
node.setPropertyValue("F_value_removal", 8.0)
# "Output..." section
node.setPropertyValue("model_fit", True)
node.setPropertyValue("r_squared_change", True)
node.setPropertyValue("selection_criteria", True)
node.setPropertyValue("descriptives", True)
node.setPropertyValue("p_correlations", True)
node.setPropertyValue("collinearity_diagnostics", True)
node.setPropertyValue("confidence_interval", True)
node.setPropertyValue("covariance_matrix", True)
node.setPropertyValue("durbin_watson", True)

```

Table 1. regressionnode properties

regressionnode Properties	Values	Property description
target	field	Regression models require a single target field and one or more input fields. A weight field can also be specified. See the topic Common modeling node properties for more information.
method	Enter Stepwise Backwards Forwards	
include_constant	flag	
use_weight	flag	
weight_field	field	
mode	Simple Expert	
complete_records	flag	
tolerance	1.0E-11 1.0E-2 1.0E-3 1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10 1.0E-11 1.0E-12	Use double quotes for arguments.
stepping_method	useP useF	useP : use probability of F useF: use F value
probability_entry	number	
probability_removal	number	
F_value_entry	number	
F_value_removal	number	
selection_criteria	flag	
confidence_interval	flag	
covariance_matrix	flag	
collinearity_diagnostics	flag	
regression_coefficients	flag	
exclude_fields	flag	

regressionnode Properties	Values	Property description
durbin_watson	flag	
model_fit	flag	
r_squared_change	flag	
p_correlations	flag	
descriptives	flag	
calculate_variable_importance	flag	

sequencenode properties



The Sequence node discovers association rules in sequential or time-oriented data. A sequence is a list of item sets that tends to occur in a predictable order. For example, a customer who purchases a razor and aftershave lotion may purchase shaving cream the next time he shops. The Sequence node is based on the CARMA association rules algorithm, which uses an efficient two-pass method for finding sequences.

Example

```
node = stream.create("sequence", "My node")
# "Fields" tab
node.setPropertyValue("id_field", "Age")
node.setPropertyValue("contiguous", True)
node.setPropertyValue("use_time_field", True)
node.setPropertyValue("time_field", "Date1")
node.setPropertyValue("content_fields", ["Drug", "BP"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Sequence_test")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("min_supp", 15.0)
node.setPropertyValue("min_conf", 14.0)
node.setPropertyValue("max_size", 7)
node.setPropertyValue("max_predictions", 5)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("use_max_duration", True)
node.setPropertyValue("max_duration", 3.0)
node.setPropertyValue("use_pruning", True)
node.setPropertyValue("pruning_value", 4.0)
node.setPropertyValue("set_mem_sequences", True)
node.setPropertyValue("mem_sequences", 5.0)
node.setPropertyValue("use_gaps", True)
node.setPropertyValue("min_item_gap", 20.0)
node.setPropertyValue("max_item_gap", 30.0)
```

Table 1. sequencenode properties

sequencenode Properties	Values	Property description
id_field	field	To create a Sequence model, you need to specify an ID field, an optional time field, and one or more content fields. Weight and frequency fields are not used. See the topic Common modeling node properties for more information.
time_field	field	
use_time_field	flag	
content_fields	[field1 ... fieldn]	
contiguous	flag	
min_supp	number	
min_conf	number	
max_size	number	
max_predictions	number	
mode	Simple Expert	

sequencenode Properties	Values	Property description
use_max_duration	flag	
max_duration	number	
use_gaps	flag	
min_item_gap	number	
max_item_gap	number	
use_pruning	flag	
pruning_value	number	
set_mem_sequences	flag	
mem_sequences	integer	

slrmnode properties

	The Self-Learning Response Model (SLRM) node enables you to build a model in which a single new case, or small number of new cases, can be used to reestimate the model without having to retrain the model using all data.
---	---

Example

```
node = stream.create("slrm", "My node")
node.setPropertyValue("target", "Offer")
node.setPropertyValue("target_response", "Response")
node.setPropertyValue("inputs", ["Cust_ID", "Age", "Ave_Bal"])
```

Table 1. slrmnode properties

slrmnode Properties	Values	Property description
target	field	The target field must be a nominal or flag field. A frequency field can also be specified. See the topic Common modeling node properties for more information.
target_response	field	Type must be flag.
continue_training_existing_model	flag	
target_field_values	flag	Use all: Use all values from source. Specify: Select values required.
target_field_values_specify	[field1 ... fieldN]	
include_model_assessment	flag	
model_assessment_random_seed	number	Must be a real number.
model_assessment_sample_size	number	Must be a real number.
model_assessment_iterations	number	Number of iterations.
display_model_evaluation	flag	
max_predictions	number	
randomization	number	
scoring_random_seed	number	
sort	Ascending Descending	Specifies whether the offers with the highest or lowest scores will be displayed first.
model_reliability	flag	
calculate_variable_importance	flag	

statisticsmodelnode properties

	The Statistics Model node enables you to analyze and work with your data by running IBM® SPSS® Statistics procedures that produce PMML. This node requires a licensed copy of IBM SPSS Statistics.
---	--

The properties for this node are described under [statisticsmodelnode properties](#).

stpnnode properties



The Spatio-Temporal Prediction (STP) node uses data that contains location data, input fields for prediction (predictors), a time field, and a target field. Each location has numerous rows in the data that represent the values of each predictor at each time of measurement. After the data is analyzed, it can be used to predict target values at any location within the shape data that is used in the analysis.

Table 1. stpnnode properties

stpnnode properties	Data type	Property description
Fields tab		
target	field	This is the target field.
location	field	The location field for the model. Only geospatial fields are allowed.
location_label	field	The categorical field to be used in the output to label the locations chosen in location
time_field	field	The time field for the model. Only fields with continuous measurement are allowed, and the storage type must be time, date, timestamp, or integer.
inputs	[field1 ... fieldN]	A list of input fields.
Time Intervals tab		
interval_type_timestamp	Years Quarters Months Weeks Days Hours Minutes Seconds	
interval_type_date	Years Quarters Months Weeks Days	
interval_type_time	Hours Minutes Seconds	Limits the number of days per week that are taken into account when creating the time index that STP uses for calculation
interval_type_integer	Periods (Time index fields only, Integer storage)	The interval to which the data set will be converted. The selection available is dependent on the storage type of the field that is chosen as the time_field for the model.
period_start	integer	
start_month	January February March April May June July August September October November December	The month the model will start to index from (for example, if set to March but the first record in the data set is January , the model will skip the first two records and start indexing at March).
week_begins_on	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	The starting point for the time index created by STP from the data
days_per_week	integer	Minimum 1, maximum 7, in increments of 1
hours_per_day	integer	The number of hours the model accounts for in a day. If this is set to 10 , the model will start indexing at the day_begins_at time and continue indexing for 10 hours, then skip to the next value matching the day_begins_at value, etc.
day_begins_at	00:00 01:00 02:00 03:00 ... 23:00	Sets the hour value that the model starts indexing from.
interval_increment	1 2 3 4 5 6 10 12 15 20 30	This increment setting is for minutes or seconds. This determines where the model creates indexes from the data. So with an increment of 30 and interval type seconds , the model will create an index from the data every 30 seconds.
data_matches_interval	Boolean	If set to N , the conversion of the data to the regular interval_type occurs before the model is built. If your data is already in the correct format, and the interval_type and any associated settings match your data, set this to Y to prevent the conversion or aggregation of your data. Setting this to Y disables all of the Aggregation controls.
agg_range_default	Sum Mean Min Max Median 1stQuartile 3rdQuartile	This determines the default aggregation method used for continuous fields. Any continuous fields which are not specifically included in the custom aggregation will be aggregated using the method specified here.
custom_agg	[[field, aggregation method], [..]] Demo: [['x5', 'FirstQuartile'], ['x4', 'Sum']]	Structured property: Script parameter: custom_agg For example: set :stpnnode.custom_agg = [[[field1 function] [field2 function]]] Where function is the aggregation function to be used with that field.
Basics tab		

stpmnode properties	Data type	Property description
<code>include_intercept</code>	<code>flag</code>	
<code>max_autoregressive_lag</code>	<code>integer</code>	Minimum 1, maximum 5, in increments of 1. This is the number of previous records required for a prediction. So if set to 5, for example, then the previous 5 records are used to create a new forecast. The number of records specified here from the build data are incorporated into the model and, therefore, the user does not need to provide the data again when scoring the model.
<code>estimation_method</code>	<code>Parametric Nonparametric</code>	The method for modeling the spatial covariance matrix
<code>parametric_model</code>	<code>Gaussian Exponential PoweredExponential</code>	Order parameter for <code>Parametric</code> spatial covariance model
<code>exponential_power</code>	<code>number</code>	Power level for <code>PoweredExponential</code> model. Minimum 1, maximum 2.
Advanced tab		
<code>max_missing_values</code>	<code>integer</code>	The maximum percentage of records with missing values allowed in the model.
<code>significance</code>	<code>number</code>	The significance level for hypotheses testing in the model build. Specifies the significance value for all the tests in STP model estimation, including two Goodness of Fit tests, effect F-tests, and coefficient t-tests.
Output tab		
<code>model_specifications</code>	<code>flag</code>	
<code>temporal_summary</code>	<code>flag</code>	
<code>location_summary</code>	<code>flag</code>	Determines whether the Location Summary table is included in the model output.
<code>model_quality</code>	<code>flag</code>	
<code>test_mean_structure</code>	<code>flag</code>	
<code>mean_structural_coefficients</code>	<code>flag</code>	
<code>autoregressive_coefficients</code>	<code>flag</code>	
<code>test_decay_rate</code>	<code>flag</code>	
<code>parametric_spatial_covariance</code>	<code>flag</code>	
<code>correlations_heat_map</code>	<code>flag</code>	
<code>correlations_map</code>	<code>flag</code>	
<code>location_clusters</code>	<code>flag</code>	
<code>similarity_threshold</code>	<code>number</code>	The threshold at which output clusters are considered similar enough to be merged into a single cluster.
<code>max_number_clusters</code>	<code>integer</code>	The upper limit for the number of clusters which can be included in the model output.
Model Options tab		
<code>use_model_name</code>	<code>flag</code>	
<code>model_name</code>	<code>string</code>	
<code>uncertainty_factor</code>	<code>number</code>	Minimum 0, maximum 100. Determines the increase in uncertainty (error) applied to predictions in the future. It is the upper and lower bound for the predictions.

svmnode properties

	The Support Vector Machine (SVM) node enables you to classify data into one of two groups without overfitting. SVM works well with wide data sets, such as those with a very large number of input fields.
---	--

Example

```

node = stream.create("svm", "My node")
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("kernel", "Polynomial")
node.setPropertyValue("gamma", 1.5)

```

Table 1. svmnode properties

svmnode Properties	Values	Property description
all_probabilities	flag	
stopping_criteria	1.0E-11 1.0E-2 1.0E-3 (default) 1.0E-4 1.0E-5 1.0E-6	Determines when to stop the optimization algorithm.
regularization	number	Also known as the C parameter.
precision	number	Used only if measurement level of target field is Continuous .
kernel	RBF(default) Polynomial Sigmoid Linear	Type of kernel function used for the transformation.
rbf_gamma	number	Used only if kernel is RBF.
gamma	number	Used only if kernel is Polynomial or Sigmoid.
bias	number	
degree	number	Used only if kernel is Polynomial.
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

tcmnode Properties



Temporal causal modeling attempts to discover key causal relationships in time series data. In temporal causal modeling, you specify a set of target series and a set of candidate inputs to those targets. The procedure then builds an autoregressive time series model for each target and includes only those inputs that have the most significant causal relationship with the target.

Table 1. tcmnode properties

tcmnode Properties	Values	Property description
custom_fields	Boolean	
dimensionlist	[dimension1 ... dimensionN]	
data_struct	Multiple Single	
metric_fields	fields	
both_target_and_input	[f1 ... fN]	
targets	[f1 ... fN]	
candidate_inputs	[f1 ... fN]	
forced_inputs	[f1 ... fN]	
use_timestamp	Timestamp Period	
input_interval	None Unknown Year Quarter Month Week Day Hour Hour_nonperiod Minute Minute_nonperiod Second Second_nonperiod	
period_field	string	
period_start_value	integer	
num_days_per_week	integer	
start_day_of_week	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	
num_hours_per_day	integer	
start_hour_of_day	integer	
timestamp_increments	integer	
cyclic_increments	integer	
cyclic_periods	list	
output_interval	None Year Quarter Month Week Day Hour Minute Second	
is_same_interval	Same Not same	
cross_hour	Boolean	

tcmnode Properties	Values	Property description
aggregate_and_distribute	list	
aggregate_default	Mean Sum Mode Min Max	
distribute_default	Mean Sum	
group_default	Mean Sum Mode Min Max	
missing_input	Linear_interp Series_mean K_mean K_meridian Linear_trend None	
k_mean_param	integer	
k_median_param	integer	
missing_value_threshold	integer	
conf_level	integer	
max_num_predictor	integer	
max_lag	integer	
epsilon	number	
threshold	integer	
is_re_est	Boolean	
num_targets	integer	
percent_targets	integer	
fields_display	list	
series_display	list	
network_graph_for_target	Boolean	
sign_level_for_target	number	
fit_and_outlier_for_target	Boolean	
sum_and_para_for_target	Boolean	
impact_diag_for_target	Boolean	
impact_diag_type_for_target	Effect Cause Both	
impact_diag_level_for_target	integer	
series_plot_for_target	Boolean	
res_plot_for_target	Boolean	
top_input_for_target	Boolean	
forecast_table_for_target	Boolean	
same_as_for_target	Boolean	
network_graph_for_series	Boolean	
sign_level_for_series	number	
fit_and_outlier_for_series	Boolean	
sum_and_para_for_series	Boolean	
impact_diagram_for_series	Boolean	
impact_diagram_type_for_series	Effect Cause Both	
impact_diagram_level_for_series	integer	
series_plot_for_series	Boolean	
residual_plot_for_series	Boolean	
forecast_table_for_series	Boolean	
outlier_root_cause_analysis	Boolean	
causal_levels	integer	
outlier_table	Interactive Pivot Both	
rmsp_error	Boolean	
bic	Boolean	

tcmnode Properties	Values	Property description
r_square	Boolean	
outliers_over_time	Boolean	
series_transformation	Boolean	
use_estimation_period	Boolean	
estimation_period	Times Observation	
observations	list	
observations_type	Latest Earliest	
observations_num	integer	
observations_exclude	integer	
extend_records_into_future	Boolean	
forecastperiods	integer	
max_num_distinct_values	integer	
display_targets	FIXEDNUMBER PERCENTAGE	
goodness_fit_measure	ROOTMEAN BIC RSQUARE	
top_input_for_series	Boolean	
aic	Boolean	
rmse	Boolean	

ts properties

	The Time Series node estimates exponential smoothing, univariate Autoregressive Integrated Moving Average (ARIMA), and multivariate ARIMA (or transfer function) models for time series data and produces forecasts of future performance. This Time Series node is similar to the previous Time Series node that was deprecated in SPSS® Modeler version 18. However, this newer Time Series node is designed to harness the power of IBM® SPSS Analytic Server to process big data, and display the resulting model in the output viewer that was added in SPSS Modeler version 17.
---	---

Table 1. ts properties

ts Properties	Values	Property description
targets	field	The Time Series node forecasts one or more targets, optionally using one or more input fields as predictors. Frequency and weight fields are not used. See the topic Common modeling node properties for more information.
candidate_inputs	[field1 ... fieldN]	Input or predictor fields used by the model.
use_period	flag	
date_time_field	field	
input_interval	None Unknown Year Quarter Month Week Day Hour Hour_nonperiod Minute Minute_nonperiod Second Second_nonperiod	
period_field	field	
period_start_value	integer	
num_days_per_week	integer	
start_day_of_week	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	
num_hours_per_day	integer	
start_hour_of_day	integer	
timestamp_increments	integer	
cyclic_increments	integer	
cyclic_periods	list	
output_interval	None Year Quarter Month Week Day Hour Minute Second	

ts Properties	Values	Property description
is_same_interval	flag	
cross_hour	flag	
aggregate_and_distribute	list	
aggregate_default	Mean Sum Mode Min Max	
distribute_default	Mean Sum	
group_default	Mean Sum Mode Min Max	
missing_imput	Linear_interp Series_mean K_mean K_median Linear_trend	
k_span_points	integer	
use_estimation_period	flag	
estimation_period	Observations Times	
date_estimation	list	Only available if you use <code>date_time_field</code>
period_estimation	list	Only available if you use <code>use_period</code>
observations_type	Latest Earliest	
observations_num	integer	
observations_exclude	integer	
method	ExpertModeler Exsmooth Arima	
expert_modeler_method	ExpertModeler Exsmooth Arima	
consider_seasonal	flag	
detect_outliers	flag	
expert_outlier_additive	flag	
expert_outlier_level_shift	flag	
expert_outlier_innovational	flag	
expert_outlier_level_shift	flag	
expert_outlier_transient	flag	
expert_outlier_seasonal_additive	flag	
expert_outlier_local_trend	flag	
expert_outlier_additive_patch	flag	
consider_new_esmodels	flag	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative DampedTrendAdditive DampedTrendMultiplicative MultiplicativeTrendAdditive MultiplicativeSeasonal MultiplicativeTrendMultiplicative MultiplicativeTrend	Specifies the Exponential Smoothing method. Default is Simple.

ts Properties	Values	Property description
futureValue_type_method	Compute specify	If Compute is used, the system computes the Future Values for the forecast period for each predictor. For each predictor, you can choose from a list of functions (blank, mean of recent points, most recent value) or use specify to enter values manually. To specify individual fields and properties, use the extend_metric_values property. For example: set :ts.futureValue_type_method="specify" set :ts.extend_metric_values=[{'Market_1','USER_SPECIFY',[1,2,3]},{'Market_2','MOST_RECENT_VALUE',''},{'Market_3','RECENT_POINTS_MEAN',''}]
exsmooth_transformation_type	None SquareRoot NaturalLog	
arima.p	integer	
arima.d	integer	
arima.q	integer	
arima.sp	integer	
arima.sd	integer	
arima.sq	integer	
arima_transformation_type	None SquareRoot NaturalLog	
arima_include_constant	flag	
tf_arima.p.fieldname	integer	For transfer functions.
tf_arima.d.fieldname	integer	For transfer functions.
tf_arima.q.fieldname	integer	For transfer functions.
tf_arima.sp.fieldname	integer	For transfer functions.
tf_arima.sd.fieldname	integer	For transfer functions.
tf_arima.sq.fieldname	integer	For transfer functions.
tf_arima.dely.fieldname	integer	For transfer functions.
tf_arima_transformation_type.fieldname	None SquareRoot NaturalLog	For transfer functions.
arima_detect_outliers	flag	
arima_outlier_additive	flag	
arima_outlier_level_shift	flag	
arima_outlier_innovational	flag	
arima_outlier_transient	flag	
arima_outlier_seasonal_additive	flag	
arima_outlier_local_trend	flag	
arima_outlier_additive_patch	flag	
max_lags	integer	
cal_PI	flag	

ts Properties	Values	Property description
<code>conf_limit_pc</code>	<code>real</code>	
<code>events</code>	<code>fields</code>	
<code>continue</code>	<code>flag</code>	
<code>scoring_mode_l_only</code>	<code>flag</code>	Use for models with very large numbers (tens of thousands) of time series.
<code>forecastperiods</code>	<code>integer</code>	
<code>extend_records_into_future</code>	<code>flag</code>	
<code>extend_metric_values</code>	<code>fields</code>	Allows you to provide future values for predictors.
<code>conf_limits</code>	<code>flag</code>	
<code>noise_res</code>	<code>flag</code>	
<code>max_models_output</code>	<code>integer</code>	Controls how many models are shown in output. Default is 10. Models are not shown in output if the total number of models built exceeds this value. Models are still available for scoring.

treeas properties

	The Tree-AS node is similar to the existing CHAID node; however, the Tree-AS node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS® Modeler version 17. The node generates a decision tree by using chi-square statistics (CHAID) to identify optimal splits. This use of CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.
---	---

Table 1. treeas properties

treeas Properties	Values	Property description
<code>target</code>	<code>field</code>	In the Tree-AS node, CHAID models require a single target and one or more input fields. A frequency field can also be specified. See the topic Common modeling node properties for more information.
<code>method</code>	<code>chaid</code> <code>exhaustive_chaid</code>	
<code>max_depth</code>	<code>integer</code>	Maximum tree depth, from 0 to 20. The default value is 5.
<code>num_bins</code>	<code>integer</code>	Only used if the data is made up of continuous inputs. Set the number of equal frequency bins to be used for the inputs; options are: 2, 4, 5, 10, 20, 25, 50, or 100.
<code>record_threshold</code>	<code>integer</code>	The number of records at which the model will switch from using p-values to Effect sizes while building the tree. The default is 1,000,000; increase or decrease this in increments of 10,000.
<code>split_alpha</code>	<code>number</code>	Significance level for splitting. The value must be between 0.01 and 0.99.
<code>merge_alpha</code>	<code>number</code>	Significance level for merging. The value must be between 0.01 and 0.99.
<code>bonferroni_adjustment</code>	<code>flag</code>	Adjust significance values using Bonferroni method.
<code>effect_size_threshold_continuous</code>	<code>number</code>	Set the Effect size threshold when splitting nodes and merging categories when using a continuous target. The value must be between 0.01 and 0.99.
<code>effect_size_threshold_categorical</code>	<code>number</code>	Set the Effect size threshold when splitting nodes and merging categories when using a categorical target. The value must be between 0.01 and 0.99.
<code>split_merged_categories</code>	<code>flag</code>	Allow resplitting of merged categories.
<code>grouping_significance_level</code>	<code>number</code>	Used to determine how groups of nodes are formed or how unusual nodes are identified.
<code>chi_square</code>	<code>pearson</code> <code>likelihood_ratio</code>	Method used to calculate the chi-square statistic: Pearson or Likelihood Ratio
<code>minimum_records_use</code>	<code>use_percentage</code> <code>use_absolute</code>	
<code>min_parent_records_pc</code>	<code>number</code>	Default value is 2. Minimum 1, maximum 100, in increments of 1. Parent branch value must be higher than child branch.
<code>min_child_records_pc</code>	<code>number</code>	Default value is 1. Minimum 1, maximum 100, in increments of 1.

treesas Properties	Values	Property description
min_parent_records_abs	number	Default value is 100. Minimum 1, maximum 100, in increments of 1. Parent branch value must be higher than child branch.
min_child_records_abs	number	Default value is 50. Minimum 1, maximum 100, in increments of 1.
epsilon	number	Minimum change in expected cell frequencies..
max_iterations	number	Maximum iterations for convergence.
use_costs	flag	
costs	structured	Structured property. The format is a list of 3 values: the actual value, the predicted value, and the cost if that prediction is wrong. For example: tree.setPropertyValue("costs", [{"drugA": "drugB", "value": 3.0}, {"drugX": "drugY", "value": 4.0}])
default_cost_increase	none linear square custom	Note: only enabled for ordinal targets. Set default values in the costs matrix.
calculate_conf	flag	
display_rule_id	flag	Adds a field in the scoring output that indicates the ID for the terminal node to which each record is assigned.

twostepnode Properties



The TwoStep node uses a two-step clustering method. The first step makes a single pass through the data to compress the raw input data into a manageable set of subclusters. The second step uses a hierarchical clustering method to progressively merge the subclusters into larger and larger clusters. TwoStep has the advantage of automatically estimating the optimal number of clusters for the training data. It can handle mixed field types and large data sets efficiently.

Example

```
node = stream.create("twostep", "My node")
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["Age", "K", "Na", "BP"])
node.setPropertyValue("partition", "Test")
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "TwoStep_Drug")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("exclude_outliers", True)
node.setPropertyValue("cluster_label", "String")
node.setPropertyValue("label_prefix", "TwoStep_")
node.setPropertyValue("cluster_num_auto", False)
node.setPropertyValue("max_num_clusters", 9)
node.setPropertyValue("min_num_clusters", 3)
node.setPropertyValue("num_clusters", 7)
```

Table 1. twostepnode properties

twostepnode Properties	Values	Property description
inputs	[field1 ... fieldN]	TwoStep models use a list of input fields, but no target. Weight and frequency fields are not recognized. See the topic Common modeling node properties for more information.
standardize	flag	
exclude_outliers	flag	
percentage	number	
cluster_num_auto	flag	
min_num_clusters	number	
max_num_clusters	number	
num_clusters	number	
cluster_label	String Number	
label_prefix	string	
distance_measure	Euclidean Loglikelihood	
clustering_criterion	AIC BIC	

twostepAS Properties

	TwoStep Cluster is an exploratory tool that is designed to reveal natural groupings (or clusters) within a data set that would otherwise not be apparent. The algorithm that is employed by this procedure has several desirable features that differentiate it from traditional clustering techniques, such as handling of categorical and continuous variables, automatic selection of number of clusters, and scalability.
---	---

Table 1. twostepAS properties

twostepAS Properties	Values	Property description
<code>inputs</code>	<code>[f1 ... fN]</code>	TwoStepAS models use a list of input fields, but no target. Weight and frequency fields are not recognized.
<code>use_predefined_roles</code>	Boolean	Default=True
<code>use_custom_field_assignments</code>	Boolean	Default=False
<code>cluster_num_auto</code>	Boolean	Default=True
<code>min_num_clusters</code>	integer	Default=2
<code>max_num_clusters</code>	integer	Default=15
<code>num_clusters</code>	integer	Default=5
<code>clustering_criterion</code>	<code>AIC BIC</code>	
<code>automatic_clustering_method</code>	<code>use_clustering_criterion_setting</code> <code>Distance_jump Minimum Maximum</code>	
<code>feature_importance_method</code>	<code>use_clustering_criterion_setting</code> <code>effect_size</code>	
<code>use_random_seed</code>	Boolean	
<code>random_seed</code>	integer	
<code>distance_measure</code>	<code>Euclidean Loglikelihood</code>	
<code>include_outlier_clusters</code>	Boolean	Default=True
<code>num_cases_in_feature_tree_leaf_is_less_than</code>	integer	Default=10
<code>top_perc_outliers</code>	integer	Default=5
<code>initial_dist_change_threshold</code>	integer	Default=0
<code>leaf_node_maximum_branches</code>	integer	Default=8
<code>non_leaf_node_maximum_branches</code>	integer	Default=8
<code>max_tree_depth</code>	integer	Default=3
<code>adjustment_weight_on_measurement_level</code>	integer	Default=6
<code>memory_allocation_mb</code>	number	Default=512
<code>delayed_split</code>	Boolean	Default=True
<code>fields_to_standardize</code>	<code>[f1 ... fN]</code>	
<code>adaptive_feature_selection</code>	Boolean	Default=True
<code>featureMisPercent</code>	integer	Default=70
<code>coefRange</code>	number	Default=0.05
<code>percCasesSingleCategory</code>	integer	Default=95
<code>numCases</code>	integer	Default=24
<code>include_model_specifications</code>	Boolean	Default=True
<code>include_record_summary</code>	Boolean	Default=True
<code>include_field_transformations</code>	Boolean	Default=True
<code>excluded_inputs</code>	Boolean	Default=True
<code>evaluate_model_quality</code>	Boolean	Default=True
<code>show_feature_importance_bar_chart</code>	Boolean	Default=True
<code>show_feature_importance_word_cloud</code>	Boolean	Default=True
<code>show_outlier_clusters_interactive_table_and_chart</code>	Boolean	Default=True
<code>show_outlier_clusters_pivot_table</code>	Boolean	Default=True
<code>across_cluster_feature_importance</code>	Boolean	Default=True
<code>across_cluster_profiles_pivot_table</code>	Boolean	Default=True

twostepAS Properties	Values	Property description
<code>withinprofiles</code>	Boolean	Default=True
<code>cluster_distances</code>	Boolean	Default=True
<code>cluster_label</code>	<code>String Number</code>	
<code>label_prefix</code>	<code>String</code>	

Model nugget node properties

Model nugget nodes share the same common properties as other nodes. See the topic [Common Node Properties](#) for more information.

- [`applyanomalydetectionnode Properties`](#)
- [`applypriorinode Properties`](#)
- [`applyassociationrulesnode Properties`](#)
- [`applyautoclassifiernode Properties`](#)
- [`applyautoclusternode Properties`](#)
- [`applyautonumericnode Properties`](#)
- [`applybayesnetnode Properties`](#)
- [`applyc50node Properties`](#)
- [`applycarmanode Properties`](#)
- [`applycartnode Properties`](#)
- [`applychaidnode Properties`](#)
- [`applycoxregnode Properties`](#)
- [`applydecisionlistnode Properties`](#)
- [`applydiscriminantnode Properties`](#)
- [`applyextension properties`](#)
- [`applyfactornode Properties`](#)
- [`applyfeatureselectionnode Properties`](#)
- [`applygeneralizedlinearnode Properties`](#)
- [`applyglmmnode Properties`](#)
- [`applygle Properties`](#)
- [`applygmm properties`](#)
- [`applykmeansnode Properties`](#)
- [`applyknnnode Properties`](#)
- [`applykohonennode Properties`](#)
- [`applylinearnode Properties`](#)
- [`applylinearasnode Properties`](#)
- [`applylogregnode Properties`](#)
- [`applylsvmnode Properties`](#)
- [`applyneuralnetnode Properties`](#)
- [`applyneuralnetworknode properties`](#)
- [`applyocsvmnode properties`](#)
- [`applyquestnode Properties`](#)
- [`applyrandomtrees Properties`](#)
- [`applyregressionnode Properties`](#)
- [`applyselflearningnode properties`](#)
- [`applysequencenode Properties`](#)
- [`applysvmnode Properties`](#)
- [`applystpnode Properties`](#)
- [`applytcminode Properties`](#)
- [`applyts Properties`](#)
- [`applytimeseriesnode Properties \(deprecated\)`](#)
- [`applytreeas Properties`](#)
- [`applytwostepnode Properties`](#)
- [`applytwostepAS Properties`](#)
- [`applyxgboosttreenode properties`](#)
- [`applyxgboostlinearnode properties`](#)
- [`hdbscan nugget properties`](#)
- [`kdeapply properties`](#)

applyanomalydetectionnode Properties

Anomaly Detection modeling nodes can be used to generate an Anomaly Detection model nugget. The scripting name of this model nugget is `applyanomalydetectionnode`. For more information on scripting the modeling node itself, [anomalydetectionnode properties](#)

Table 1. `applyanomalydetectionnode` properties

<code>applyanomalydetectionnode Properties</code>	Values	Property description
<code>anomaly_score_method</code>	<code>FlagAndScore</code> <code>FlagOnly</code> <code>ScoreOnly</code>	Determines which outputs are created for scoring.
<code>num_fields</code>	<code>integer</code>	Fields to report.
<code>discard_records</code>	<code>flag</code>	Indicates whether records are discarded from the output or not.
<code>discard_anomalous_records</code>	<code>flag</code>	Indicator of whether to discard the anomalous or <i>non</i> -anomalous records. The default is <code>off</code> , meaning that <i>non</i> -anomalous records are discarded. Otherwise, if <code>on</code> , anomalous records will be discarded. This property is enabled only if the <code>discard_records</code> property is enabled.

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarminode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyapriorinode Properties

Apriori modeling nodes can be used to generate an Apriori model nugget. The scripting name of this model nugget is `applyapriorinode`. For more information on scripting the modeling node itself, [apriorinode properties](#)

Table 1. `applyapriorinode` properties

<code>applyapriorinode Properties</code>	Values	Property description
<code>max_predictions</code>	<code>number (integer)</code>	
<code>ignore_unmatched</code>	<code>flag</code>	
<code>allow_repeats</code>	<code>flag</code>	
<code>check_basket</code>	<code>NoPredictions Predictions NoCheck</code>	
<code>criterion</code>	<code>Confidence Support RuleSupport Lift Deployability</code>	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmenode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyassociationrulesnode Properties

The Association Rules modeling node can be used to generate an association rules model nugget. The scripting name of this model nugget is `applyassociationrulesnode`. For more information on scripting the modeling node itself, see [associationrulesnode.properties](#).

Table 1. applyassociationrulesnode properties

<code>applyassociationrulesnode properties</code>	Data type	Property description
<code>max_predictions</code>	<code>integer</code>	The maximum number of rules that can be applied to each input to the score.
<code>criterion</code>	<code>Confidence Rule support Lift Conditions support Deployability</code>	Select the measure used to determine the strength of rules.
<code>allow_repeats</code>	<code>Boolean</code>	Determine whether rules with the same prediction are included in the score.
<code>check_input</code>	<code>NoPredictions Predictions NoCheck</code>	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmenode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)

- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applystpnode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyautoclassifiernode Properties

Auto Classifier modeling nodes can be used to generate an Auto Classifier model nugget. The scripting name of this model nugget is `applyautoclassifiernode`. For more information on scripting the modeling node itself, [autoclassifiernode.properties](#)

Table 1. applyautoclassifiernode properties

applyautoclassifiernode Properties	Values	Property description
<code>flag_ensemble_method</code>	<code>Voting</code> <code>Evaluation</code> <code>WeightedVoting</code> <code>Confidence</code> <code>WeightedVoting</code> <code>RawPropensity</code> <code>WeightedVoting</code> <code>HighestConfidence</code> <code>AverageRawPropensity</code>	Specifies the method used to determine the ensemble score. This setting applies only if the selected target is a flag field.
<code>flag_evaluation_selection</code>	<code>Accuracy</code> <code>AUC_ROC</code>	This option is for flag target only, to decide which evaluation measure is chosen for evaluation-weighted voting.
<code>filter_individual_model_output</code>	<code>flag</code>	Specifies whether scoring results from individual models should be suppressed.
<code>is_ensemble_update</code>	<code>flag</code>	Enables continuous auto machine learning mode, which adds new component models into an existing auto model set instead of replacing the existing auto model, and re-evaluates measures of existing component models using newly available data.
<code>is_auto_ensemble_weights_reevaluation</code>	<code>flag</code>	Enables automatic model weights reevaluation.
<code>use_accumulated_factor</code>	<code>flag</code>	Accumulated factor is used to compute accumulated measures.
<code>accumulated_factor</code>	<code>number (double)</code>	Max value is <code>0 . 99</code> , and min value is <code>0 . 85</code> .
<code>use_accumulated_reducing</code>	<code>flag</code>	Performs model reducing based on accumulated limit during model refresh.
<code>accumulated_reducing_limit</code>	<code>number (double)</code>	Max value is <code>0 . 7</code> , and min value is <code>0 . 1</code> .
<code>use_accumulated_weighted_evaluation</code>	<code>flag</code>	Accumulated evaluation measure is used for voting when the evaluation-weighted voting method is selected for the ensemble method.
<code>flag_voting_tie_selection</code>	<code>Random</code> <code>HighestConfidence</code> <code>RawPropensity</code>	If a voting method is selected, specifies how ties are resolved. This setting applies only if the selected target is a flag field.
<code>set_ensemble_method</code>	<code>Voting</code> <code>Evaluation</code> <code>WeightedVoting</code> <code>Confidence</code> <code>WeightedVoting</code> <code>HighestConfidence</code>	Specifies the method used to determine the ensemble score. This setting applies only if the selected target is a set field.

applyautoclusternode Properties	Values	Property description
<code>set_voting_type_selection</code>	<code>Random</code> <code>HighestConfidence</code>	If a voting method is selected, specifies how ties are resolved. This setting applies only if the selected target is a nominal field.

applyautoclusternode Properties

Auto Cluster modeling nodes can be used to generate an Auto Cluster model nugget. The scripting name of this model nugget is `applyautoclusternode`. No other properties exist for this model nugget. For more information on scripting the modeling node itself, [autoclusternode properties](#)

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmenode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnnnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyautonumericnode Properties

Auto Numeric modeling nodes can be used to generate an Auto Numeric model nugget. The scripting name of this model nugget is `applyautonumericnode`. For more information on scripting the modeling node itself, [autonumericnode properties](#)

Table 1. applyautonumericnode properties

applyautonumericnode Properties	Values	Property description
<code>calculate_standard_error</code>	<code>flag</code>	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)

- [applyautoclassifiernode Properties](#)
 - [applyautoclusternode Properties](#)
 - [applybayesnetnode Properties](#)
 - [applyc50node Properties](#)
 - [applycarmannode Properties](#)
 - [applycartnode Properties](#)
 - [applychaidnode Properties](#)
 - [applycoxregnode Properties](#)
 - [applydecisionlistnode Properties](#)
 - [applydiscriminantnode Properties](#)
 - [applyfactornode Properties](#)
 - [applyfeatureselectionnode Properties](#)
 - [applygeneralizedlinearnode Properties](#)
 - [applyglmnode Properties](#)
 - [applykmeansnode Properties](#)
 - [applyknnnode Properties](#)
 - [applykohonennode Properties](#)
 - [applylinearnode Properties](#)
 - [applylogregnode Properties](#)
 - [applyneuralnetnode Properties](#)
 - [applyquestnode Properties](#)
 - [applyregressionnode Properties](#)
 - [applyselflearningnode_properties](#)
 - [applysequencenode Properties](#)
 - [applysvmnode Properties](#)
 - [applytimeseriesnode Properties \(deprecated\)](#)
 - [applytwostepnode Properties](#)
-

applybayesnetnode Properties

Bayesian network modeling nodes can be used to generate a Bayesian network model nugget. The scripting name of this model nugget is `applybayesnetnode`. For more information on scripting the modeling node itself, [bayesnetnode properties](#).

Table 1. applybayesnetnode properties

applybayesnetnode Properties	Values	Property description
<code>all_probabilities</code>	<code>flag</code>	
<code>raw_propensity</code>	<code>flag</code>	
<code>adjusted_propensity</code>	<code>flag</code>	
<code>calculate_raw_propensities</code>	<code>flag</code>	
<code>calculate_adjusted_propensities</code>	<code>flag</code>	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applyc50node Properties](#)
- [applycarmannode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)

- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyc50node Properties

C5.0 modeling nodes can be used to generate a C5.0 model nugget. The scripting name of this model nugget is *applyc50node*. For more information on scripting the modeling node itself, [c50node properties](#).

Table 1. applyc50node properties

applyc50node Properties	Values	Property description
<code>sql_generate</code>	<code>udf Never NoMissingValues</code>	Used to set SQL generation options during rule set execution. The default value is <code>udf</code> .
<code>calculate_conf</code>	<code>flag</code>	Available when SQL generation is enabled; this property includes confidence calculations in the generated tree.
<code>calculate_raw_propensities</code>	<code>flag</code>	
<code>calculate_adjusted_propensities</code>	<code>flag</code>	

applycarmanode Properties

CARMA modeling nodes can be used to generate a CARMA model nugget. The scripting name of this model nugget is *applycarmanode*. No other properties exist for this model nugget. For more information on scripting the modeling node itself, [carmanode properties](#).

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)

- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applycartnode Properties

C&R Tree modeling nodes can be used to generate a C&R Tree model nugget. The scripting name of this model nugget is *applycartnode*. For more information on scripting the modeling node itself, [cartnode properties](#).

Table 1. applycartnode properties

applycartnode Properties	Values	Property description
enable_sql_generation	Never MissingValues NoMissingValues	Used to set SQL generation options during rule set execution.
calculate_conf	flag	Available when SQL generation is enabled; this property includes confidence calculations in the generated tree.
display_rule_id	flag	Adds a field in the scoring output that indicates the ID for the terminal node to which each record is assigned.
calculate_raw_propensities	flag	
calculate_adjusted_probabilities	flag	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmanode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applychaidnode Properties

CHAID modeling nodes can be used to generate a CHAID model nugget. The scripting name of this model nugget is *applychaidnode*. For more information on scripting the modeling node itself, [chaidnode properties](#).

Table 1. applychaidnode properties

applychaidnode Properties	Values	Property description
enable_sql_generation	Never MissingValues	Used to set SQL generation options during rule set execution.
calculate_conf	flag	
display_rule_id	flag	Adds a field in the scoring output that indicates the ID for the terminal node to which each record is assigned.
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmanode Properties](#)
- [applycartnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applycoxregnode Properties

Cox modeling nodes can be used to generate a Cox model nugget. The scripting name of this model nugget is `applycoxregnode`. For more information on scripting the modeling node itself, [coxregnode properties](#).

Table 1. applycoxregnode properties

applycoxregnode Properties	Values	Property description
future_time_as	Intervals Fields	
time_interval	number	
num_future_times	integer	
time_field	field	
past_survival_time	field	
all_probabilities	flag	
cumulative_hazard	flag	

Related information

- [Node properties overview](#)

- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmancode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogrnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applydecisionlistnode Properties

Decision List modeling nodes can be used to generate a Decision List model nugget. The scripting name of this model nugget is `applydecisionlistnode`. For more information on scripting the modeling node itself, [decisionlistnode properties](#).

Table 1. applydecisionlistnode properties

applydecisionlistnode Properties	Values	Property description
<code>enable_sql_generation</code>	<code>flag</code>	When true, IBM® SPSS® Modeler will try to push back the Decision List model to SQL.
<code>calculate_raw_propensities</code>	<code>flag</code>	
<code>calculate_adjusted_propensities</code>	<code>flag</code>	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmancode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)

- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applydiscriminantnode Properties

Discriminant modeling nodes can be used to generate a Discriminant model nugget. The scripting name of this model nugget is `applydiscriminantnode`. For more information on scripting the modeling node itself, [discriminantnode properties](#).

Table 1. applydiscriminantnode properties

applydiscriminantnode Properties	Values	Property description
<code>calculate_raw_propensities</code>	<code>flag</code>	
<code>calculate_adjusted_propensities</code>	<code>flag</code>	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmannode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyextension properties

	Extension Model nodes can be used to generate an Extension model nugget. The scripting name of this model nugget is
--	---



applyextension. For more information on scripting the modeling node itself, see [extensionmodelnode properties](#).

Python for Spark example

```
#### script example for Python for Spark
applyModel = stream.findByType("extension_apply", None)

score_script = """
import json
import spss.pyspark.runtime
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.linalg import DenseVector
from pyspark.mllib.tree import DecisionTreeModel
from pyspark.sql.types import StringType, StructField

cxt = spss.pyspark.runtime.getContext()

if cxt.isComputeDataModelOnly():
    _schema = cxt.getSparkInputSchema()
    _schema.fields.append(StructField("Prediction", StringType(), nullable=True))
    cxt.setSparkOutputSchema(_schema)
else:
    df = cxt.getSparkInputData()

    _modelPath = cxt.getModelContentToPath("TreeModel")
    metadata = json.loads(cxt.getModelContentToString("model.dm"))

    schema = df.dtypes[:]
    target = "Drug"
    predictors = ["Age", "BP", "Sex", "Cholesterol", "Na", "K"]

    lookup = {}
    for i in range(0,len(schema)):
        lookup[schema[i][0]] = i

    def row2LabeledPoint(dm,lookup,target,predictors,row):
        target_index = lookup[target]
        tval = dm[target_index].index(row[target_index])
        pvals = []
        for predictor in predictors:
            predictor_index = lookup[predictor]
            if isinstance(dm[predictor_index],list):
                pval = row[predictor_index] in dm[predictor_index] and
dm[predictor_index].index(row[predictor_index]) or -1
            else:
                pval = row[predictor_index]
            pvals.append(pval)
        return LabeledPoint(tval, DenseVector(pvals))

    # convert dataframe to an RDD containing LabeledPoint
    lps = df.rdd.map(lambda row: row2LabeledPoint(metadata,lookup,target,predictors,row))
    treeModel = DecisionTreeModel.load(cxt.getSparkContext(), _modelPath);
    # score the model, produces an RDD containing just double values
    predictions = treeModel.predict(lps.map(lambda lp: lp.features))

    def addPrediction(x,dm,lookup,target):
        result = []
        for _idx in range(0, len(x[0])):
            result.append(x[0][_idx])
        result.append(dm[lookup[target]][int(x[1])])
        return result

        _schema = cxt.getSparkInputSchema()
        _schema.fields.append(StructField("Prediction", StringType(), nullable=True))
        rdd2 = df.rdd.zip(predictions).map(lambda x:addPrediction(x, metadata, lookup, target))
        outDF = cxt.getSparkSQLContext().createDataFrame(rdd2, _schema)

        cxt.setSparkOutputData(outDF)
"""

applyModel.setPropertyValue("python_syntax", score_script)
```

R example

```
#### script example for R
applyModel.setPropertyValue("r_syntax", """
result<-predict(modelerModel,newdata=modelerData)
```

```

modelerData<-cbind(modelerData,result)
var1<-c(fieldName="NaPrediction",fieldLabel="",fieldStorage="real",fieldMeasure="",
fieldFormat="",fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)"""

```

Table 1. applyextension properties

applyextension Properties	Values	Property Description
<code>r_syntax</code>	<code>string</code>	R scripting syntax for model scoring.
<code>python_syntax</code>	<code>string</code>	Python scripting syntax for model scoring.
<code>use_batch_size</code>	<code>flag</code>	Enable use of batch processing.
<code>batch_size</code>	<code>integer</code>	Specify the number of data records to be included in each batch.
<code>convert_flags</code>	<code>StringsAndDouble s LogicalValues</code>	Option to convert flag fields.
<code>convert_missing</code>	<code>flag</code>	Option to convert missing values to the R NA value.
<code>convert_datetime</code>	<code>flag</code>	Option to convert variables with date or datetime formats to R date/time formats.
<code>convert_datetime_class</code>	<code>POSIXct POSIXlt</code>	Options to specify to what format variables with date or datetime formats are converted.

applyfactornode Properties

PCA/Factor modeling nodes can be used to generate a PCA/Factor model nugget. The scripting name of this model nugget is `applyfactornode`. No other properties exist for this model nugget. For more information on scripting the modeling node itself, [factornode properties](#).

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmannode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyfeatureselectionnode Properties

Feature Selection modeling nodes can be used to generate a Feature Selection model nugget. The scripting name of this model nugget is `applyfeatureselectionnode`. For more information on scripting the modeling node itself, [featureselectionnode properties](#).

Table 1. `applyfeatureselectionnode` properties

<code>applyfeatureselectionnode Properties</code>	Values	Property description
<code>selected_ranked_fields</code>		Specifies which ranked fields are checked in the model browser.
<code>selected_screened_fields</code>		Specifies which screened fields are checked in the model browser.

Related information

- [Node properties overview](#)
- [Model nugget node.properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmenode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode.properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applygeneralizedlinearnode Properties

Generalized Linear (genlin) modeling nodes can be used to generate a Generalized Linear model nugget. The scripting name of this model nugget is `applygeneralizedlinearnode`. For more information on scripting the modeling node itself, [genlinnode properties](#).

Table 1. `applygeneralizedlinearnode` properties

<code>applygeneralizedlinearnode Properties</code>	Values	Property description
<code>calculate_raw_propensities</code>	<code>flag</code>	
<code>calculate_adjusted_propensities</code>	<code>flag</code>	

Related information

- [Node properties overview](#)
- [Model nugget node.properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)

- [applycarmannode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyglmmnode Properties

GLMM modeling nodes can be used to generate a GLMM model nugget. The scripting name of this model nugget is *applyglmmnode*. For more information on scripting the modeling node itself, [glmmnode properties](#).

Table 1. applyglmmnode properties

applyglmmnode Properties	Values	Property description
confidence	onProbability onIncrease	Basis for computing scoring confidence value: highest predicted probability, or difference between highest and second highest predicted probabilities.
score_categorical_probabilities	flag	If set to True , produces the predicted probabilities for categorical targets. A field is created for each category. Default is False .
max_categories	integer	Maximum number of categories for which to predict probabilities. Used only if score_category_probabilities is True .
score_propensity	flag	If set to True , produces raw propensity scores (likelihood of "True" outcome) for models with flag targets. If partitions are in effect, also produces adjusted propensity scores based on the testing partition. Default is False .
enable_sql_generation	udf native	Used to set SQL generation options during stream execution. The options are to pushback to the database and score using a SPSS® Modeler Server scoring adapter (if connected to a database with a scoring adapter installed), or to score within SPSS Modeler. The default value is udf .

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmannode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)

- [applygeneralizedlinearnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applygle Properties

The GLE modeling node can be used to generate a GLE model nugget. The scripting name of this model nugget is *applygle*. For more information on scripting the modeling node itself, see [gle properties](#).

Table 1. applygle properties

applygle Properties	Values	Property description
<code>enable_sql_generation</code>	<code>udf native</code>	Used to set SQL generation options during stream execution. Choose either to pushback to the database and score using a SPSS® Modeler Server scoring adapter (if connected to a database with a scoring adapter installed), or score within SPSS Modeler.

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmanode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applygmm properties

The Gaussian Mixture node can be used to generate a Gaussian Mixture model nugget. The scripting name of this model nugget is *applygmm*. The properties in the following table are available in version 18.2.1.1 and later. For more information on scripting the modeling node itself, see [gmm properties](#).

Table 1. *applygmm* properties

applygmm properties	Data type	Property description
centers		
item_count		
total		
dimension		
components		
partition		

applykmeansnode Properties

K-Means modeling nodes can be used to generate a K-Means model nugget. The scripting name of this model nugget is *applykmeansnode*. No other properties exist for this model nugget. For more information on scripting the modeling node itself, [kmeansnode properties](#).

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmannode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyknnnode Properties

KNN modeling nodes can be used to generate a KNN model nugget. The scripting name of this model nugget is *applyknnnode*. For more information on scripting the modeling node itself, [knnnode properties](#).

Table 1. *applyknnnode* properties

applyknnnode Properties	Values	Property description
all_probabilities	<i>flag</i>	
save_distances	<i>flag</i>	

Related information

- [Node properties overview](#)
 - [Model nugget node properties](#)
 - [applyanomalydetectionnode Properties](#)
 - [applyapriorinode Properties](#)
 - [applyautoclassifiernode Properties](#)
 - [applyautoclusternode Properties](#)
 - [applyautonumericnode Properties](#)
 - [applybayesnetnode Properties](#)
 - [applyc50node Properties](#)
 - [applycarmanode Properties](#)
 - [applycartnode Properties](#)
 - [applychaidnode Properties](#)
 - [applycoxregnode Properties](#)
 - [applydecisionlistnode Properties](#)
 - [applydiscriminantnode Properties](#)
 - [applyfactornode Properties](#)
 - [applyfeatureselectionnode Properties](#)
 - [applygeneralizedlinearnode Properties](#)
 - [applygimnnode Properties](#)
 - [applykmeansnode Properties](#)
 - [applykohonennode Properties](#)
 - [applylinearnode Properties](#)
 - [applylogregnode Properties](#)
 - [applyneuralnetnode Properties](#)
 - [applyquestnode Properties](#)
 - [applyregressionnode Properties](#)
 - [applyselflearningnode_properties](#)
 - [applysequencenode Properties](#)
 - [applysvmnode Properties](#)
 - [applytimeseriesnode Properties \(deprecated\)](#)
 - [applytwostepnode Properties](#)
-

applykohonennode Properties

Kohonen modeling nodes can be used to generate a Kohonen model nugget. The scripting name of this model nugget is `applykohonennode`. No other properties exist for this model nugget. For more information on scripting the modeling node itself, [c50node.properties](#).

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmanode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applygimnnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)

- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applylinearnode Properties

Linear modeling nodes can be used to generate a Linear model nugget. The scripting name of this model nugget is *applylinearnode*. For more information on scripting the modeling node itself, [linernode properties](#).

Table 1. applylinearnode Properties

linear Properties	Values	Property description
<code>use_custom_name</code>	<code>flag</code>	
<code>custom_name</code>	<code>string</code>	
<code>enable_sql_generation</code>	<code>udf native puresql</code>	Used to set SQL generation options during stream execution. The options are to pushback to the database and score using a SPSS® Modeler Server scoring adapter (if connected to a database with a scoring adapter installed), to score within SPSS Modeler, or to pushback to the database and score using SQL. The default value is <code>udf</code> .

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmanode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode.properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applylinearasnode Properties

Linear-AS modeling nodes can be used to generate a Linear-AS model nugget. The scripting name of this model nugget is `applylinearasnode`. For more information on scripting the modeling node itself, [linearasnode properties](#).

Table 1. `applylinearasnode` Properties

<code>applylinearasnode</code> Property	Values	Property description
<code>enable_sql_generation</code>	<code>udf native</code>	The default value is <code>udf</code> .

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmannode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applylogregnode Properties

Logistic Regression modeling nodes can be used to generate a Logistic Regression model nugget. The scripting name of this model nugget is `applylogregnode`. For more information on scripting the modeling node itself, [logregnode properties](#).

Table 1. `applylogregnode` properties

<code>applylogregnode</code> Properties	Values	Property description
<code>calculate_raw_propensities</code>	<code>flag</code>	
<code>calculate_conf</code>	<code>flag</code>	
<code>enable_sql_generation</code>	<code>flag</code>	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)

- [applycarmenode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applysvmnode Properties

LSVM modeling nodes can be used to generate an LSVM model nugget. The scripting name of this model nugget is `applysvmnode`. For more information on scripting the modeling node itself, see [svmnode properties](#).

Table 1. applysvmnode properties

applysvmnode Properties	Values	Property description
<code>calculate_raw_propensities</code>	<code>flag</code>	Specifies whether to calculate raw propensity scores.
<code>enable_sql_generation</code>	<code>udf native</code>	Specifies whether to score using the Scoring Adapter (if installed) or in process, or to score outside of the database.

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmenode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)

- [applysequencenode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyneuralnetnode Properties

Neural Net modeling nodes can be used to generate a Neural Net model nugget. The scripting name of this model nugget is *applyneuralnetnode*. For more information on scripting the modeling node itself, [neuralnetnode properties](#).

Caution: A newer version of the Neural Net nugget, with enhanced features, is available in this release and is described in the next section (*applyneuralnetwork*). Although the previous version is still available, we recommend updating your scripts to use the new version. Details of the previous version are retained here for reference, but support for it will be removed in a future release.

Table 1. applyneuralnetnode properties

applyneuralnetnode Properties	Values	Property description
calculate_conf	flag	Available when SQL generation is enabled; this property includes confidence calculations in the generated tree.
enable_sql_generation	flag	
nn_score_method	Difference SoftMax	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	

Related information

- [Node properties overview](#)
- [Model nugget node.properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmanode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyneuralnetworknode properties

Neural Network modeling nodes can be used to generate a Neural Network model nugget. The scripting name of this model nugget is *applyneuralnetworknode*. For more information on scripting the modeling node itself, [neuralnetworknode Properties](#).

Table 1. applyneuralnetworknode properties

applyneuralnetworknode Properties	Values	Property description
<code>use_custom_name</code>	<code>flag</code>	
<code>custom_name</code>	<code>string</code>	
<code>confidence</code>	<code>onProbability onIncrease</code>	
<code>score_categorical_probabilities</code>	<code>flag</code>	
<code>max_categories</code>	<code>number</code>	
<code>score_propriety</code>	<code>flag</code>	
<code>enable_sql_generation</code>	<code>udf native puresql</code>	Used to set SQL generation options during stream execution. The options are to pushback to the database and score using a SPSS® Modeler Server scoring adapter (if connected to a database with a scoring adapter installed), to score within SPSS Modeler, or to pushback to the database and score using SQL. The default value is <code>udf</code> .

applyocsvmnode properties

One-Class SVM nodes can be used to generate a One-Class SVM model nugget. The scripting name of this model nugget is *applyocsvmnode*. No other properties exist for this model nugget. For more information on scripting the modeling node itself, see [ocsvmnode properties](#).

applyquestnode Properties

QUEST modeling nodes can be used to generate a QUEST model nugget. The scripting name of this model nugget is *applyquestnode*. For more information on scripting the modeling node itself, [questnode properties](#).

Table 1. applyquestnode properties

applyquestnode Properties	Values	Property description
<code>enable_sql_generation</code>	<code>Never MissingValues</code> <code>NoMissingValues</code>	Used to set SQL generation options during rule set execution.
<code>calculate_conf</code>	<code>flag</code>	
<code>display_rule_id</code>	<code>flag</code>	Adds a field in the scoring output that indicates the ID for the terminal node to which each record is assigned.
<code>calculate_raw_propensities</code>	<code>flag</code>	
<code>calculate_adjusted_propensities</code>	<code>flag</code>	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applypriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmanode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)

- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyrandomtrees Properties

The Random Trees modeling node can be used to generate a Random Trees model nugget. The scripting name of this model nugget is `applyrandomtrees`. For more information on scripting the modeling node itself, see [randomtrees properties](#).

Table 1. applyrandomtrees properties

<code>applyrandomtrees Properties</code>	Values	Property description
<code>calculate_conf</code>	<code>flag</code>	This property includes confidence calculations in the generated tree.
<code>enable_sql_generation</code>	<code>udf native</code>	Used to set SQL generation options during stream execution. Choose either to pushback to the database and score using a SPSS® Modeler Server scoring adapter (if connected to a database with a scoring adapter installed), or score within SPSS Modeler.

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmannode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)

- [applytwostepnode Properties](#)

applyregressionnode Properties

Linear Regression modeling nodes can be used to generate a Linear Regression model nugget. The scripting name of this model nugget is `applyregressionnode`. No other properties exist for this model nugget. For more information on scripting the modeling node itself, [regressionnode properties](#).

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmenode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applyselflearningnode properties

Self-Learning Response Model (SLRM) modeling nodes can be used to generate a SLRM model nugget. The scripting name of this model nugget is `applyselflearningnode`. For more information on scripting the modeling node itself, [slrmnode properties](#).

Table 1. applyselflearningnode properties

applyselflearningnode Properties	Values	Property description
<code>max_predictions</code>	<code>number</code>	
<code>randomization</code>	<code>number</code>	
<code>scoring_random_seed</code>	<code>number</code>	
<code>sort</code>	<code>ascending</code> <code>descending</code>	Specifies whether the offers with the highest or lowest scores will be displayed first.
<code>model_reliability</code>	<code>flag</code>	Takes account of model reliability option on Settings tab.

applysequencenode Properties

Sequence modeling nodes can be used to generate a Sequence model nugget. The scripting name of this model nugget is *applysequencenode*. No other properties exist for this model nugget. For more information on scripting the modeling node itself, [sequencenode properties](#).

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmancode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applysvmnode Properties

SVM modeling nodes can be used to generate an SVM model nugget. The scripting name of this model nugget is *applysvmnode*. For more information on scripting the modeling node itself, [svmnode properties](#).

Table 1. applysvmnode properties

applysvmnode Properties	Values	Property description
<code>all_probabilities</code>	<code>flag</code>	
<code>calculate_raw_propensities</code>	<code>flag</code>	
<code>calculate_adjusted_propensities</code>	<code>flag</code>	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmancode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)

- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applystpnode Properties

The STP modeling node can be used to generate an associated model nugget, which display the model output in the Output Viewer. The scripting name of this model nugget is `applystpnode`. For more information on scripting the modeling node itself, see [stpnode.properties](#).

Table 1. applystpnode properties

applystpnode properties	Data type	Property description
<code>uncertainty_factor</code>	<code>Boolean</code>	Minimum 0, maximum 100.

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmanode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnode Properties](#)
- [applyassociationrulesnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applytcmnode Properties

Temporal Causal Modeling (TCM) modeling nodes can be used to generate a TCM model nugget. The scripting name of this model nugget is `applytcmnode`. For more information on scripting the modeling node itself, see [tcmnode Properties](#).

Table 1. applytcmnode properties

applytcmnode Properties	Values	Property description
<code>ext_future</code>	<code>boolean</code>	
<code>ext_future_num</code>	<code>integer</code>	
<code>noise_res</code>	<code>boolean</code>	
<code>conf_limits</code>	<code>boolean</code>	
<code>target_fields</code>	<code>list</code>	
<code>target_series</code>	<code>list</code>	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmannode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytwostepnode Properties](#)

applyts Properties

The Time Series modeling node can be used to generate a Time Series model nugget. The scripting name of this model nugget is `applyts`. For more information on scripting the modeling node itself, see [ts_properties](#).

Table 1. applyts properties

applyts Properties	Values	Property description
<code>extend_records_into_future</code>	<code>Boolean</code>	
<code>ext_future_num</code>	<code>integer</code>	
<code>compute_future_values_input</code>	<code>Boolean</code>	
<code>forecastperiods</code>	<code>integer</code>	
<code>noise_res</code>	<code>boolean</code>	

applyts Properties	Values	Property description
conf_limits	boolean	
target_fields	list	
target_series	list	
includeTargets	field	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmancode Properties](#)
- [applycartnode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytwostepnode Properties](#)

applytimeseriesnode Properties (deprecated)

The Time Series modeling node can be used to generate a Time Series model nugget. The scripting name of this model nugget is `applytimeseriesnode`. For more information on scripting the modeling node itself, [timeseriesnode properties \(deprecated\)](#).

Table 1. applytimeseriesnode properties

applytimeseriesnode Properties	Values	Property description
calculate_conf	flag	
calculate_residuals	flag	

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmancode Properties](#)
- [applycartnode Properties](#)

- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytwostepnode Properties](#)

applytreeas Properties

Tree-AS modeling nodes can be used to generate a Tree-AS model nugget. The scripting name of this model nugget is *applytreeas*. For more information on scripting the modeling node itself, see [treeas properties](#).

Table 1. applytreeas properties

applytreeas Properties	Values	Property description
<code>calculate_conf</code>	<code>flag</code>	This property includes confidence calculations in the generated tree.
<code>display_rule_id</code>	<code>flag</code>	Adds a field in the scoring output that indicates the ID for the terminal node to which each record is assigned.
<code>enable_sql_generation</code>	<code>udf native</code>	Used to set SQL generation options during stream execution. Choose either to pushback to the database and score using a SPSS® Modeler Server scoring adapter (if connected to a database with a scoring adapter installed), or score within SPSS Modeler.

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarminode Properties](#)
- [applychaidnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)

- [applyselflearningnode properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)
- [applytwostepnode Properties](#)

applytwostepnode Properties

TwoStep modeling nodes can be used to generate a TwoStep model nugget. The scripting name of this model nugget is *applytwostepnode*. No other properties exist for this model nugget. For more information on scripting the modeling node itself, [twostepnode Properties](#).

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)
- [applyanomalydetectionnode Properties](#)
- [applyapriorinode Properties](#)
- [applyautoclassifiernode Properties](#)
- [applyautoclusternode Properties](#)
- [applyautonumericnode Properties](#)
- [applybayesnetnode Properties](#)
- [applyc50node Properties](#)
- [applycarmenode Properties](#)
- [applycartnode Properties](#)
- [applychainnode Properties](#)
- [applycoxregnode Properties](#)
- [applydecisionlistnode Properties](#)
- [applydiscriminantnode Properties](#)
- [applyfactornode Properties](#)
- [applyfeatureselectionnode Properties](#)
- [applygeneralizedlinearnode Properties](#)
- [applyglmnode Properties](#)
- [applykmeansnode Properties](#)
- [applyknnnode Properties](#)
- [applykohonennode Properties](#)
- [applylinearnode Properties](#)
- [applylogregnode Properties](#)
- [applyneuralnetnode Properties](#)
- [applyquestnode Properties](#)
- [applyregressionnode Properties](#)
- [applyselflearningnode_properties](#)
- [applysequencenode Properties](#)
- [applysvmnode Properties](#)
- [applytimeseriesnode Properties \(deprecated\)](#)

applytwostepAS Properties

TwoStep AS modeling nodes can be used to generate a TwoStep AS model nugget. The scripting name of this model nugget is *applytwostepAS*. For more information on scripting the modeling node itself, [twostepAS Properties](#).

Table 1. applytwostepAS Properties

applytwostep AS Properties	Values	Property description
enable_sql_generation	udf native	Used to set SQL generation options during stream execution. The options are to pushback to the database and score using a SPSS® Modeler Server scoring adapter (if connected to a database with a scoring adapter installed), or to score within SPSS Modeler. The default value is udf .

Related information

- [Node properties overview](#)
- [Model nugget node properties](#)

- [applyanomalydetectionnode Properties](#)
 - [applyapriorinode Properties](#)
 - [applyautoclassifiernode Properties](#)
 - [applyautoclusternode Properties](#)
 - [applyautonumericnode Properties](#)
 - [applybayesnetnode Properties](#)
 - [applyc50node Properties](#)
 - [applycarmannode Properties](#)
 - [applycartnode Properties](#)
 - [applychaidnode Properties](#)
 - [applycoxregnode Properties](#)
 - [applydecisionlistnode Properties](#)
 - [applydiscriminantnode Properties](#)
 - [applyfactornode Properties](#)
 - [applyfeatureselectionnode Properties](#)
 - [applygeneralizedlinearnode Properties](#)
 - [applyglmmnode Properties](#)
 - [applykmeansnode Properties](#)
 - [applyknnnode Properties](#)
 - [applykohonennode Properties](#)
 - [applylinearnode Properties](#)
 - [applylogregnode Properties](#)
 - [applyneuralnetnode Properties](#)
 - [applyquestnode Properties](#)
 - [applyregressionnode Properties](#)
 - [applyselflearningnode_properties](#)
 - [applysequencenode Properties](#)
 - [applysvmnode Properties](#)
 - [applytimeseriesnode Properties \(deprecated\)](#)
-

applyxgboosttreenode properties

The XGBoost Tree node can be used to generate an XGBoost Tree model nugget. The scripting name of this model nugget is `applyxgboosttreenode`. The properties in the following table were added in 18.2.1.1. For more information on scripting the modeling node itself, see [xgboosttreenode Properties](#).

Table 1. applyxgboosttreenode properties

<code>applyxgboosttreenode properties</code>	Data type	Property description
<code>use_model_name</code>		
<code>model_name</code>		

applyxgboostlinearnode properties

XGBoost Linear nodes can be used to generate an XGBoost Linear model nugget. The scripting name of this model nugget is `applyxgboostlinearnode`. No other properties exist for this model nugget. For more information on scripting the modeling node itself, see [xgboostlinearnode Properties](#).

hdbscannugget properties

The HDBSCAN node can be used to generate an HDBSCAN model nugget. The scripting name of this model nugget is `hdbscannugget`. No other properties exist for this model nugget. For more information on scripting the modeling node itself, see [hdbscannode properties](#).

kdeapply properties

The KDE Modeling node can be used to generate a KDE model nugget. The scripting name of this model nugget is `kdeapply`. For information on scripting the modeling node itself, see [kdemodel properties](#).

Table 1. kdeapply properties

kdeapply properties	Data type	Property description
<code>outLogDensity</code> Renamed to <code>out_log_density</code> starting with version 18.2.1.1	<code>boolean</code>	Specify <code>True</code> or <code>False</code> to include or exclude the log density value in the output. Default is <code>False</code> .

Database modeling node properties

IBM® SPSS® Modeler supports integration with data mining and modeling tools available from database vendors, including Microsoft SQL Server Analysis Services, Oracle Data Mining, and IBM Netezza® Analytics. You can build and score models using native database algorithms, all from within the IBM SPSS Modeler application. Database models can also be created and manipulated through scripting using the properties described in this section.

For example, the following script excerpt illustrates the creation of a Microsoft Decision Trees model by using the IBM SPSS Modeler scripting interface:

```
stream = modeler.script.stream()
msbuilder = stream.createAt("mstreenode", "MSBuilder", 200, 200)

msbuilder.setPropertyValue("analysis_server_name", 'localhost')
msbuilder.setPropertyValue("analysis_database_name", 'TESTDB')
msbuilder.setPropertyValue("mode", 'Expert')
msbuilder.setPropertyValue("datasource", 'LocalServer')
msbuilder.setPropertyValue("target", 'Drug')
msbuilder.setPropertyValue("inputs", ['Age', 'Sex'])
msbuilder.setPropertyValue("unique_field", 'IDX')
msbuilder.setPropertyValue("custom_fields", True)
msbuilder.setPropertyValue("model_name", 'MSDRUG')

typenode = stream.findByType("type", None)
stream.link(typenode, msbuilder)
results = []
msbuilder.run(results)
msapplier = stream.createModelApplierAt(results[0], "Drug", 200, 300)
tablenode = stream.createAt("table", "Results", 300, 300)
stream.linkBetween(msapplier, typenode, tablenode)
msapplier.setPropertyValue("sql_generate", True)
tablenode.run([])
```

- [Node Properties for Microsoft Modeling](#)
- [Node Properties for Oracle Modeling](#)
- [Node Properties for IBM Netezza Analytics Modeling](#)

Node Properties for Microsoft Modeling

- [Microsoft Modeling Node Properties](#)
- [Microsoft Model Nugget Properties](#)

Microsoft Modeling Node Properties

Common Properties

The following properties are common to the Microsoft database modeling nodes.

Table 1. Common Microsoft node properties

Common Microsoft Node Properties	Values	Property Description
<code>analysis_database_name</code>	<code>string</code>	Name of the Analysis Services database.
<code>analysis_server_name</code>	<code>string</code>	Name of the Analysis Services host.
<code>use_transactional_data</code>	<code>flag</code>	Specifies whether input data is in tabular or transactional format.
<code>inputs</code>	<code>list</code>	Input fields for tabular data.
<code>target</code>	<code>field</code>	Predicted field (not applicable to MS Clustering or Sequence Clustering nodes).
<code>unique_field</code>	<code>field</code>	Key field.
<code>msas_parameters</code>	<code>structured</code>	Algorithm parameters. See the topic Algorithm Parameters for more information.

Common Microsoft Node Properties	Values	Property Description
with_drillthrough	flag	With Drillthrough option.

MS Decision Tree

There are no specific properties defined for nodes of type `mstreenode`. See the common Microsoft properties at the start of this section.

MS Clustering

There are no specific properties defined for nodes of type `msclusternode`. See the common Microsoft properties at the start of this section.

MS Association Rules

The following specific properties are available for nodes of type `msassocnode`:

Table 2. `msassocnode` properties

msassocnode Properties	Values	Property Description
<code>id_field</code>	<code>field</code>	Identifies each transaction in the data.
<code>trans_inputs</code>	<code>list</code>	Input fields for transactional data.
<code>transactional_target</code>	<code>field</code>	Predicted field (transactional data).

MS Naive Bayes

There are no specific properties defined for nodes of type `msbayesnode`. See the common Microsoft properties at the start of this section.

MS Linear Regression

There are no specific properties defined for nodes of type `msregressionnode`. See the common Microsoft properties at the start of this section.

MS Neural Network

There are no specific properties defined for nodes of type `msneuralnetworknode`. See the common Microsoft properties at the start of this section.

MS Logistic Regression

There are no specific properties defined for nodes of type `mslogisticnode`. See the common Microsoft properties at the start of this section.

MS Time Series

There are no specific properties defined for nodes of type `mstimeseriesnode`. See the common Microsoft properties at the start of this section.

MS Sequence Clustering

The following specific properties are available for nodes of type `mssequenceclusternode`:

Table 3. `mssequenceclusternode` properties

mssequenceclusternode Properties	Values	Property Description
<code>id_field</code>	<code>field</code>	Identifies each transaction in the data.
<code>input_fields</code>	<code>list</code>	Input fields for transactional data.
<code>sequence_field</code>	<code>field</code>	Sequence identifier.
<code>target_field</code>	<code>field</code>	Predicted field (tabular data).

- [Algorithm Parameters](#)

Algorithm Parameters

Each Microsoft database model type has specific parameters that can be set using the `msas_parameters` property--for example:

```

stream = modeler.script.stream()
msregressionnode = stream.findByType("msregression", None)
msregressionnode.setPropertyValue("msas_parameters", [[ "MAXIMUM_INPUT_ATTRIBUTES", 255], ["MAXIMUM_OUTPUT_ATTRIBUTES", 255]])

```

These parameters are derived from SQL Server. To see the relevant parameters for each node:

1. Place a database source node on the canvas.
2. Open the database source node.
3. Select a valid source from the Data source drop-down list.
4. Select a valid table from the Table name list.
5. Click OK to close the database source node.
6. Attach the Microsoft database modeling node whose properties you want to list.
7. Open the database modeling node.
8. Select the Expert tab.

The available **msas_parameters** properties for this node are displayed.

Microsoft Model Nugget Properties

The following properties are for the model nuggets created using the Microsoft database modeling nodes.

MS Decision Tree

Table 1. MS Decision Tree properties

applymsstreenode Properties	Values	Description
analysis_database_name	<i>string</i>	This node can be scored directly in a stream. This property is used to identify the name of the Analysis Services database.
analysis_server_name	<i>string</i>	Name of the Analysis server host.
datasource	<i>string</i>	Name of the SQL Server ODBC data source name (DSN).
sql_generate	<i>flag udf</i>	Enables SQL generation.

MS Linear Regression

Table 2. MS Linear Regression properties

applymsregressionnode Properties	Values	Description
analysis_database_name	<i>string</i>	This node can be scored directly in a stream. This property is used to identify the name of the Analysis Services database.
analysis_server_name	<i>string</i>	Name of the Analysis server host.

MS Neural Network

Table 3. MS Neural Network properties

applymsneuralnetworknode Properties	Values	Description
analysis_database_name	<i>string</i>	This node can be scored directly in a stream. This property is used to identify the name of the Analysis Services database.
analysis_server_name	<i>string</i>	Name of the Analysis server host.

MS Logistic Regression

Table 4. MS Logistic Regression properties

applymslogisticnode Properties	Values	Description
analysis_database_name	<i>string</i>	This node can be scored directly in a stream. This property is used to identify the name of the Analysis Services database.
analysis_server_name	<i>string</i>	Name of the Analysis server host.

MS Time Series

Table 5. MS Time Series properties

Properties	Values	Description
<code>analysis_database_name</code>	<code>string</code>	This node can be scored directly in a stream. This property is used to identify the name of the Analysis Services database.
<code>analysis_server_name</code>	<code>string</code>	Name of the Analysis server host.
<code>start_from</code>	<code>new_prediction</code> <code>historical_prediction</code>	Specifies whether to make future predictions or historical predictions.
<code>new_step</code>	<code>number</code>	Defines starting time period for future predictions.
<code>historical_step</code>	<code>number</code>	Defines starting time period for historical predictions.
<code>end_step</code>	<code>number</code>	Defines ending time period for predictions.

MS Sequence Clustering

Table 6. MS Sequence Clustering properties

Properties	Values	Description
<code>analysis_database_name</code>	<code>string</code>	This node can be scored directly in a stream. This property is used to identify the name of the Analysis Services database.
<code>analysis_server_name</code>	<code>string</code>	Name of the Analysis server host.

Node Properties for Oracle Modeling

- [Oracle Modeling Node Properties](#)
- [Oracle Model Nugget Properties](#)

Oracle Modeling Node Properties

The following properties are common to Oracle database modeling nodes.

Table 1. Common Oracle node properties

Common Oracle Node Properties	Values	Property Description
<code>target</code>	<code>field</code>	
<code>inputs</code>	<code>List of fields</code>	
<code>partition</code>	<code>field</code>	Field used to partition the data into separate samples for the training, testing, and validation stages of model building.
<code>datasource</code>		
<code>username</code>		
<code>password</code>		
<code>epassword</code>		
<code>use_model_name</code>	<code>flag</code>	
<code>model_name</code>	<code>string</code>	Custom name for new model.
<code>use_partitioned_data</code>	<code>flag</code>	If a partition field is defined, this option ensures that only data from the training partition is used to build the model.
<code>unique_field</code>	<code>field</code>	
<code>auto_data_prep</code>	<code>flag</code>	Enables or disables the Oracle automatic data preparation feature (11g databases only).
<code>costs</code>	<code>structured</code>	Structured property in the form: <code>[[drugA drugB 1.5] [drugA drugC 2.1]]</code> , where the arguments in <code>[]</code> are actual predicted costs.
<code>mode</code>	<code>Simple</code> <code>Expert</code>	Causes certain properties to be ignored if set to <code>Simple</code> , as noted in the individual node properties.
<code>use_prediction_probability</code>	<code>flag</code>	

Common Oracle Node Properties	Values	Property Description
<code>prediction_probability</code>	<code>string</code>	
<code>use_prediction_set</code>	<code>flag</code>	

Oracle Naive Bayes

The following properties are available for nodes of type `oranbnode`.

Table 2. oranbnode properties

oranbnode Properties	Values	Property Description
<code>singleton_threshold</code>	<code>number</code>	0.0–1.0.*
<code>pairwise_threshold</code>	<code>number</code>	0.0–1.0.*
<code>priors</code>	<code>Data Equal Custom</code>	
<code>custom_priors</code>	<code>structured</code>	Structured property in the form: <code>set :oranbnode.custom_priors = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]</code>

* Property ignored if `mode` is set to `Simple`.

Oracle Adaptive Bayes

The following properties are available for nodes of type `oraabnnnode`.

Table 3. oraabnnnode properties

oraabnnnode Properties	Values	Property Description
<code>model_type</code>	<code>SingleFeature</code> <code>MultiFeature NaiveBayes</code>	
<code>use_execution_time_limit</code>	<code>flag</code>	*
<code>execution_time_limit</code>	<code>integer</code>	Value must be greater than 0.*
<code>max_naive_bayes_predictors</code>	<code>integer</code>	Value must be greater than 0.*
<code>max_predictors</code>	<code>integer</code>	Value must be greater than 0.*
<code>priors</code>	<code>Data Equal Custom</code>	
<code>custom_priors</code>	<code>structured</code>	Structured property in the form: <code>set :oraabnnnode.custom_priors = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]</code>

* Property ignored if `mode` is set to `Simple`.

Oracle Support Vector Machines

The following properties are available for nodes of type `orasvmnode`.

Table 4. orasvmnode properties

orasvmnode Properties	Values	Property Description
<code>active_learning</code>	<code>Enable Disable</code>	
<code>kernel_function</code>	<code>Linear Gaussian System</code>	
<code>normalization_method</code>	<code>zscore minmax none</code>	
<code>kernel_cache_size</code>	<code>integer</code>	Gaussian kernel only. Value must be greater than 0.*
<code>convergence_tolerance</code>	<code>number</code>	Value must be greater than 0.*
<code>use_standard_deviation</code>	<code>flag</code>	Gaussian kernel only.*
<code>standard_deviation</code>	<code>number</code>	Value must be greater than 0.*
<code>use_epsilon</code>	<code>flag</code>	Regression models only.*
<code>epsilon</code>	<code>number</code>	Value must be greater than 0.*

orasvmnode Properties	Values	Property Description
use_complexity_factor	flag	*
complexity_fact or	number	*
use_outlier_rate	flag	One-Class variant only.*
outlier_rate	number	One-Class variant only. 0.0–1.0.*
weights	Data Equal Custom	
custom_weights	structured	Structured property in the form: <code>set :orasvmnode.custom_weights = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]</code>

* Property ignored if mode is set to Simple.

Oracle Generalized Linear Models

The following properties are available for nodes of type oraglmnode.

Table 5. oraglmnode properties

oraglmnode Properties	Values	Property Description
normalization_method	zscore minmax none	
missing_value_handling	ReplaceWithMean UseCompleteRecords	
use_row_weights	flag	*
row_weights_field	field	*
save_row_diagnostics	flag	*
row_diagnostics_table	string	*
coefficient_confidence	number	*
use_reference_category	flag	*
reference_category	string	*
ridge_regression	Auto Off On	*
parameter_value	number	*
vif_for_ridge	flag	*

* Property ignored if mode is set to Simple.

Oracle Decision Tree

The following properties are available for nodes of type oradecisiontreenode.

Table 6. oradecisiontreenode properties

oradecisiontreenode Properties	Values	Property Description
use_costs	flag	
impurity_metric	Entropy Gini	
term_max_depth	integer	2–20.*
term_minpct_node	number	0.0–10.0.*
term_minpct_split	number	0.0–20.0.*
term_minrec_node	integer	Value must be greater than 0.*
term_minrec_split	integer	Value must be greater than 0.*
display_rule_ids	flag	*

* Property ignored if mode is set to Simple.

Oracle O-Cluster

The following properties are available for nodes of type oraoclusternode.

Table 7. oraoclusternode properties

oraoclusternode Properties	Values	Property Description
max_num_clusters	integer	Value must be greater than 0.
max_buffer	integer	Value must be greater than 0.*
sensitivity	number	0.0–1.0.*

* Property ignored if mode is set to Simple.

Oracle KMeans

The following properties are available for nodes of type `orakmeansnode`.

Table 8. orakmeansnode properties

orakmeansnode Properties	Values	Property Description
<code>num_clusters</code>	<code>integer</code>	Value must be greater than 0.
<code>normalization_method</code>	<code>zscore minmax none</code>	
<code>distance_function</code>	<code>Euclidean Cosine</code>	
<code>iterations</code>	<code>integer</code>	0–20.*
<code>conv_tolerance</code>	<code>number</code>	0.0–0.5.*
<code>split_criterion</code>	<code>Variance Size</code>	Default is Variance.*
<code>num_bins</code>	<code>integer</code>	Value must be greater than 0.*
<code>block_growth</code>	<code>integer</code>	1–5.*
<code>min_pct_attr_support</code>	<code>number</code>	0.0–1.0.*

* Property ignored if `mode` is set to `Simple`.

Oracle NMF

The following properties are available for nodes of type `oranmfnode`.

Table 9. oranmfnode properties

oranmfnode Properties	Values	Property Description
<code>normalization_method</code>	<code>minmax none</code>	
<code>use_num_features</code>	<code>flag</code>	*
<code>num_features</code>	<code>integer</code>	0–1. Default value is estimated from the data by the algorithm.*
<code>random_seed</code>	<code>number</code>	*
<code>num_iterations</code>	<code>integer</code>	0–500.*
<code>conv_tolerance</code>	<code>number</code>	0.0–0.5.*
<code>display_all_features</code>	<code>flag</code>	*

* Property ignored if `mode` is set to `Simple`.

Oracle Apriori

The following properties are available for nodes of type `oraapriorinode`.

Table 10. oraapriorinode properties

oraapriorinode Properties	Values	Property Description
<code>content_field</code>	<code>field</code>	
<code>id_field</code>	<code>field</code>	
<code>max_rule_length</code>	<code>integer</code>	2–20.
<code>min_confidence</code>	<code>number</code>	0.0–1.0.
<code>min_support</code>	<code>number</code>	0.0–1.0.
<code>use_transactional_data</code>	<code>flag</code>	

Oracle Minimum Description Length (MDL)

There are no specific properties defined for nodes of type `oramdlnode`. See the common Oracle properties at the start of this section.

Oracle Attribute Importance (AI)

The following properties are available for nodes of type `oraainode`.

Table 11. oraainode properties

oraainode Properties	Values	Property Description
<code>custom_fields</code>	<code>flag</code>	If true, allows you to specify target, input, and other fields for the current node. If false, the current settings from an upstream Type node are used.
<code>selection_mode</code>	<code>ImportanceLevel</code> <code>ImportanceValue</code> <code>TopN</code>	

oraainode Properties	Values	Property Description
select_important	flag	When selection_mode is set to ImportanceLevel , specifies whether to select important fields.
important_label	string	Specifies the label for the "important" ranking.
select_marginal	flag	When selection_mode is set to ImportanceLevel , specifies whether to select marginal fields.
marginal_label	string	Specifies the label for the "marginal" ranking.
important_above	number	0.0–1.0.
select_unimportant	flag	When selection_mode is set to ImportanceLevel , specifies whether to select unimportant fields.
unimportant_label	string	Specifies the label for the "unimportant" ranking.
unimportant_below	number	0.0–1.0.
importance_value	number	When selection_mode is set to ImportanceValue , specifies the cutoff value to use. Accepts values from 0 to 100.
top_n	number	When selection_mode is set to TopN , specifies the cutoff value to use. Accepts values from 0 to 1000.

Oracle Model Nugget Properties

The following properties are for the model nuggets created using the Oracle models.

Oracle Naive Bayes

There are no specific properties defined for nodes of type `applyorabnnode`.

Oracle Adaptive Bayes

There are no specific properties defined for nodes of type `applyoraabnnode`.

Oracle Support Vector Machines

There are no specific properties defined for nodes of type `applyorasvmnode`.

Oracle Decision Tree

The following properties are available for nodes of type `applyoradecisiontreenode`.

Table 1. `applyoradecisiontreenode` properties

applyoradecisiontreenode Properties	Values	Property Description
use_costs	flag	
display_rule_ids	flag	

Oracle O-Cluster

There are no specific properties defined for nodes of type `applyoraoclusternode`.

Oracle KMeans

There are no specific properties defined for nodes of type `applyorakmeansnode`.

Oracle NMF

The following property is available for nodes of type `applyoranmfnode`:

Table 2. `applyoranmfnode` properties

applyoranmfnode Properties	Values	Property Description
display_all_features	flag	

Oracle Apriori

This model nugget cannot be applied in scripting.

Oracle MDL

This model nugget cannot be applied in scripting.

Node Properties for IBM® Netezza® Analytics Modeling

- [Netezza Modeling Node Properties](#)
- [Netezza Model Nugget Properties](#)

Netezza Modeling Node Properties

The following properties are common to IBM Netezza database modeling nodes.

Table 1. Common Netezza node properties

Common Netezza Node Properties	Values	Property Description
custom_fields	flag	If true, allows you to specify target, input, and other fields for the current node. If false, the current settings from an upstream Type node are used.
inputs	[field1 ... fieldN]	Input or predictor fields used by the model.
target	field	Target field (continuous or categorical).
record_id	field	Field to be used as unique record identifier.
use_upstream_connection	flag	If true (default), the connection details specified in an upstream node. Not used if move_data_to_connection is specified.
move_data_connection	flag	If true, moves the data to the database specified by connection. Not used if use_upstream_connection is specified.
connection	structured	The connection string for the Netezza database where the model is stored. Structured property in the form: ['odbc' '<dsn>' '<username>' '<psw>' '<catname>' '<conn_attribs>' [true false]] where: <dsn> is the data source name <username> and <psw> are the username and password for the database <catname> is the catalog name <conn_attribs> are the connection attributes true false indicates whether the password is needed.
table_name	string	Name of database table where model is to be stored.
use_model_name	flag	If true, uses the name specified by model_name as the name of the model, otherwise model name is created by the system.
model_name	string	Custom name for new model.
include_input_fields	flag	If true, passes all input fields downstream, otherwise passes only record_id and fields generated by model.

Netezza Decision Tree

The following properties are available for nodes of type `netezzadectreenode`.

Table 2. netezzadectreenode properties

netezzadectreenode Properties	Values	Property Description
impurity_measure	Entropy Gini	The measurement of impurity, used to evaluate the best place to split the tree.
max_tree_depth	integer	Maximum number of levels to which tree can grow. Default is 62 (the maximum possible).
min_improvement_splits	number	Minimum improvement in impurity for split to occur. Default is 0.01.

netezzadectr eenode Properties	Values	Property Description
min_instances_split	<i>integer</i>	Minimum number of unsplit records remaining before split can occur. Default is 2 (the minimum possible).
weights	<i>structured</i>	Relative weightings for classes. Structured property in the form: <code>set :netezza_dectree.weights = [[drugA 0.3] [drugB 0.6]]</code> Default is weight of 1 for all classes.
pruning_measure	Acc wAcc	Default is Acc (accuracy). Alternative wAcc (weighted accuracy) takes class weights into account while applying pruning.
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	Default is to use allTrainingData to estimate model accuracy. Use partitionTrainingData to specify a percentage of training data to use, or useOtherTable to use a training data set from a specified database table.
perc_training_data	<i>number</i>	If prune_tree_options is set to partitionTrainingData , specifies percentage of data to use for training.
prune_seed	<i>integer</i>	Random seed to be used for replicating analysis results when prune_tree_options is set to partitionTrainingData ; default is 1.
pruning_table	<i>string</i>	Table name of a separate pruning dataset for estimating model accuracy.
compute_probabilities	<i>flag</i>	If true, produces a confidence level (probability) field as well as the prediction field.

Netezza K-Means

The following properties are available for nodes of type **netezzakmeansnode**.

Table 3. netezzakmeansnode properties

netezzakmeansnode Properties	Values	Property Description
distance_measure	Euclidean Manhattan Canberra maximum	Method to be used for measuring distance between data points.
num_clusters	<i>integer</i>	Number of clusters to be created; default is 3.
max_iterations	<i>integer</i>	Number of algorithm iterations after which to stop model training; default is 5.
rand_seed	<i>integer</i>	Random seed to be used for replicating analysis results; default is 12345.

Netezza Bayes Net

The following properties are available for nodes of type **netezzabayesnode**.

Table 4. netezzabayesnode properties

netezzabayesnode Properties	Values	Property Description
base_index	<i>integer</i>	Numeric identifier assigned to first input field for internal management; default is 777.
sample_size	<i>integer</i>	Size of sample to take if number of attributes is very large; default is 10,000.
display_additional_information	<i>flag</i>	If true, displays additional progress information in a message dialog box.
type_of_prediction	best neighbors nn-neighbors	Type of prediction algorithm to use: best (most correlated neighbor), neighbors (weighted prediction of neighbors), or nn-neighbors (non null-neighbors).

Netezza Naive Bayes

The following properties are available for nodes of type **netezzanaivebayesnode**.

Table 5. netezzanaivebayesnode properties

netezzanaivebayesnode Properties	Values	Property Description
compute_probabilities	<i>flag</i>	If true, produces a confidence level (probability) field as well as the prediction field.
use_m_estimation	<i>flag</i>	If true, uses m-estimation technique for avoiding zero probabilities during estimation.

Netezza KNN

The following properties are available for nodes of type `netezzaknnnode`.

Table 6. netezzaknnnode properties

<code>netezzaknnnode</code> Properties	Values	Property Description
<code>weights</code>	<code>structured</code>	Structured property used to assign weights to individual classes. Example: <code>set :netezzaknnnode.weights = [[drugA 0.3] [drugB 0.6]]</code>
<code>distance_measure</code>	<code>Euclidean Manhattan</code> <code>Canberra Maximum</code>	Method to be used for measuring the distance between data points.
<code>num_nearest_neighbors</code>	<code>integer</code>	Number of nearest neighbors for a particular case; default is 3.
<code>standardize_measurements</code>	<code>flag</code>	If true, standardizes measurements for continuous input fields before calculating distance values.
<code>use_coresets</code>	<code>flag</code>	If true, uses core set sampling to speed up calculation for large data sets.

Netezza Divisive Clustering

The following properties are available for nodes of type `netezzadivclusternode`.

Table 7. netezzadivclusternode properties

<code>netezzadivclusternode</code> Properties	Values	Property Description
<code>distance_measure</code>	<code>Euclidean Manhattan</code> <code>Canberra Maximum</code>	Method to be used for measuring the distance between data points.
<code>max_iterations</code>	<code>integer</code>	Maximum number of algorithm iterations to perform before model training stops; default is 5.
<code>max_tree_depth</code>	<code>integer</code>	Maximum number of levels to which data set can be subdivided; default is 3.
<code>rand_seed</code>	<code>integer</code>	Random seed, used to replicate analyses; default is 12345.
<code>min_instances_split</code>	<code>integer</code>	Minimum number of records that can be split, default is 5.
<code>level</code>	<code>integer</code>	Hierarchy level to which records are to be scored; default is -1.

Netezza PCA

The following properties are available for nodes of type `netezzapcanode`.

Table 8. netezzapcanode properties

<code>netezzapcanode</code> Properties	Values	Property Description
<code>center_data</code>	<code>flag</code>	If true (default), performs data centering (also known as "mean subtraction") before the analysis.
<code>perform_data_scaling</code>	<code>flag</code>	If true, performs data scaling before the analysis. Doing so can make the analysis less arbitrary when different variables are measured in different units.
<code>force_eigensolve</code>	<code>flag</code>	If true, uses less accurate but faster method of finding principal components.
<code>pc_number</code>	<code>integer</code>	Number of principal components to which data set is to be reduced; default is 1.

Netezza Regression Tree

The following properties are available for nodes of type `netezzaregtreenode`.

Table 9. netezzaregtreenode properties

<code>netezzaregtreenode</code> Properties	Values	Property Description
<code>max_tree_depth</code>	<code>integer</code>	Maximum number of levels to which the tree can grow below the root node; default is 10.
<code>split_evaluation_measure</code>	<code>Variance</code>	Class impurity measure, used to evaluate the best place to split the tree; default (and currently only option) is <code>Variance</code> .
<code>min_improvement_splits</code>	<code>number</code>	Minimum amount to reduce impurity before new split is created in tree.
<code>min_instances_split</code>	<code>integer</code>	Minimum number of records that can be split.

netezzaregtr eenode Properties	Values	Property Description
pruning_measure	mse r2 pearson spearman	Method to be used for pruning.
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	Default is to use <code>allTrainingData</code> to estimate model accuracy. Use <code>partitionTrainingData</code> to specify a percentage of training data to use, or <code>useOtherTable</code> to use a training data set from a specified database table.
perc_training_data	number	If <code>prune_tree_options</code> is set to <code>PercTrainingData</code> , specifies percentage of data to use for training.
prune_seed	integer	Random seed to be used for replicating analysis results when <code>prune_tree_options</code> is set to <code>PercTrainingData</code> ; default is 1.
pruning_table	string	Table name of a separate pruning dataset for estimating model accuracy.
compute_probabilities	flag	If true, specifies that variances of assigned classes should be included in output.

Netezza Linear Regression

The following properties are available for nodes of type `netezzalineregressionnode`.

Table 10. netezzalineregressionnode properties

netezzalineregressionnode Properties	Values	Property Description
use_svd	flag	If true, uses Singular Value Decomposition matrix instead of original matrix, for increased speed and numerical accuracy.
include_intercept	flag	If true (default), increases overall accuracy of solution.
calculate_model_diagnoses	flag	If true, calculates diagnostics on the model.

Netezza Time Series

The following properties are available for nodes of type `netezzatimeseriesnode`.

Table 11. netezzatimeseriesnode properties

netezzatimeseriesnode Properties	Values	Property Description
time_points	field	Input field containing the date or time values for the time series.
time_series_ids	field	Input field containing time series IDs; used if input contains more than one time series.
model_table	field	Name of database table where Netezza time series model will be stored.
description_table	field	Name of input table that contains time series names and descriptions.
seasonal_adjustment_table	field	Name of output table where seasonally adjusted values computed by exponential smoothing or seasonal trend decomposition algorithms will be stored.
algorithm_name	SpectralAnalysis or spectral ExponentialSmoothing or esmoothing ARIMA SeasonalTrendDecomposition or std	Algorithm to be used for time series modeling.
trend_name	N A D A M D M	Trend type for exponential smoothing: N - none A - additive D - damped additive M - multiplicative D - damped multiplicative
seasonality_type	N A M	Seasonality type for exponential smoothing: N - none A - additive M - multiplicative
interpolation_method	linear cubic spline exponential spline	Interpolation method to be used.
timerange_setting	SD SP	Setting for time range to use: SD - system-determined (uses full range of time series data) SP - user-specified via <code>earliest_time</code> and <code>latest_time</code>
earliest_time	integer date time timestamp	Start and end values, if <code>timerange_setting</code> is <code>SP</code> . Format should follow <code>time_points</code> value. For example, if the <code>time_points</code> field contains a date, this should also be a date. Example: <code>set NZ_DT1.timerange_setting = 'SP' .set NZ_DT1.earliest_time = '1921-01-01' .set NZ_DT1.latest_time = '2121-01-01'</code>
latest_time		

netezzatimeseriesnode	Values	Property Description
arima_setting	SD SP	Setting for the ARIMA algorithm (used only if algorithm_name is set to ARIMA): SD - system-determined SP - user-specified If arima_setting = SP , use the following parameters to set the seasonal and non-seasonal values. Example (non-seasonal only): <code>set NZ_DT1.algorithm_name = 'arima' set NZ_DT1.arima_setting = 'SP' set NZ_DT1.p_symbol = 'lesseq' set NZ_DT1.p = '4' set NZ_DT1.d_symbol = 'lesseq' set NZ_DT1.d = '2' set NZ_DT1.q_symbol = 'lesseq' set NZ_DT1.q = '4'</code>
p_symbol d_symbol q_symbol sp_symbol sd_symbol sq_symbol	less eq lesseq	ARIMA - operator for parameters p , d , q , sp , sd , and sq : less - less than eq - equals lesseq - less than or equal to
p	<i>integer</i>	ARIMA - non-seasonal degrees of autocorrelation.
q	<i>integer</i>	ARIMA - non-seasonal derivation value.
d	<i>integer</i>	ARIMA - non-seasonal number of moving average orders in the model.
sp	<i>integer</i>	ARIMA - seasonal degrees of autocorrelation.
sq	<i>integer</i>	ARIMA - seasonal derivation value.
sd	<i>integer</i>	ARIMA - seasonal number of moving average orders in the model.
advanced_setting	SD SP	Determines how advanced settings are to be handled: SD - system-determined SP - user-specified via period , units_period and forecast_setting . Example: <code>set NZ_DT1.advanced_setting = 'SP' set NZ_DT1.period = 5 set NZ_DT1.units_period = 'd'</code>
period	<i>integer</i>	Length of seasonal cycle, specified in conjunction with units_period . Not applicable for spectral analysis.
units_period	ms s min h d wk q y	Units in which period is expressed: ms - milliseconds s - seconds min - minutes h - hours d - days wk - weeks q - quarters y - years For example, for a weekly time series use 1 for period and wk for units_period .
forecast_setting	forecasthorizon forecasttimes	Specifies how forecasts are to be made.
forecast_horizon	<i>integer date time timestamp</i>	If forecast_setting = forecasthorizon , specifies end point value for forecasting. Format should follow time_points value. For example, if the time_points field contains a date, this should also be a date.
forecast_times	<i>integer date time timestamp</i>	If forecast_setting = forecasttimes , specifies values to use for making forecasts. Format should follow time_points value. For example, if the time_points field contains a date, this should also be a date.
include_history	<i>flag</i>	Indicates if historical values are to be included in output.
include_interpolated_values	<i>flag</i>	Indicates if interpolated values are to be included in output. Not applicable if include_history is false .

Netezza Generalized Linear

The following properties are available for nodes of type **netezzaglmnode**.

Table 12. netezzaglmnode properties

netezzaglmnode Properties	Values	Property Description
dist_family	bernoulli gaussian poisson negativebinomial wald gamma	Distribution type; default is bernoulli .
dist_params	<i>number</i>	Distribution parameter value to use. Only applicable if distribution is Negativebinomial .
trials	<i>integer</i>	Only applicable if distribution is Binomial . When target response is a number of events occurring in a set of trials, target field contains number of events, and trials field contains number of trials.
model_table	<i>field</i>	Name of database table where Netezza generalized linear model will be stored.
maxit	<i>integer</i>	Maximum number of iterations the algorithm should perform; default is 20.
eps	<i>number</i>	Maximum error value (in scientific notation) at which algorithm should stop finding best fit model. Default is -3, meaning 1E-3, or 0.001.

netezzaglmnode Properties	Values	Property Description
tol	<i>number</i>	Value (in scientific notation) below which errors are treated as having a value of zero. Default is -7, meaning that error values below 1E-7 (or 0.0000001) are counted as insignificant.
link_func	<code>identity inverse invnegative invsquare sqrt power oddspower log clog loglog cloglog logit probit gaussit cauchit canbinom cangeom cannegbinom</code>	Link function to use; default is <code>logit</code> .
link_params	<i>number</i>	Link function parameter value to use. Only applicable if <code>link_function</code> is <code>power</code> or <code>oddspower</code> .
interaction	<code>[[[colnames1],[levels1]], [[colnames2],[levels2]], ...,[[colnamesN],[levelsN]],]</code>	Specifies interactions between fields. <code>colnames</code> is a list of input fields, and <code>level</code> is always 0 for each field. Example: <code>[[["K", "BP", "Sex", "K"], [0, 0, 0, 0]], [[("Age", "Na"), [0, 0]]]</code>
intercept	<i>flag</i>	If <code>true</code> , includes the intercept in the model.

Netezza Model Nugget Properties

The following properties are common to Netezza database model nuggets.

Table 1. Common Netezza model nugget properties

Common Netezza Model Nugget Properties	Values	Property Description
connection	<i>string</i>	The connection string for the Netezza database where the model is stored.
table_name	<i>string</i>	Name of database table where model is stored.

Other model nugget properties are the same as those for the corresponding modeling node.

The script names of the model nuggets are as follows.

Table 2. Script names of Netezza model nuggets

Model Nugget	Script Name
Decision Tree	<code>applynetezzadectreenode</code>
K-Means	<code>applynetezzakmeansnode</code>
Bayes Net	<code>applynetezzabayesnode</code>
Naive Bayes	<code>applynetezzanaivebayesnode</code>
KNN	<code>applynetezzaknnnode</code>
Divisive Clustering	<code>applynetezzadivclusternode</code>
PCA	<code>applynetezzapcanode</code>
Regression Tree	<code>applynetezzaregtreenode</code>
Linear Regression	<code>applynetezzalineregressionnode</code>
Time Series	<code>applynetezzatimeseriesnode</code>
Generalized Linear	<code>applynetezzaglmnode</code>

Output node properties

Output node properties differ slightly from those of other node types. Rather than referring to a particular node option, output node properties store a reference to the output object. This is useful in taking a value from a table and then setting it as a stream parameter.

The following topics describe the scripting properties available for output nodes.

- [analysisnode properties](#)
- [dataauditnode properties](#)
- [extensionoutputnode properties](#)
- [kdeexport properties](#)
- [matrixnode properties](#)
- [meansnode properties](#)
- [reportnode properties](#)
- [setglobalsnode properties](#)
- [simevalnode properties](#)
- [simfitnode properties](#)

- [statisticsnode properties](#)
- [statisticoutputnode Properties](#)
- [tablenode properties](#)
- [transformnode properties](#)

analysisnode properties



The Analysis node evaluates predictive models' ability to generate accurate predictions. Analysis nodes perform various comparisons between predicted values and actual values for one or more model nuggets. They can also compare predictive models to each other.

Example

```
node = stream.create("analysis", "My node")
# "Analysis" tab
node.setPropertyValue("coincidence", True)
node.setPropertyValue("performance", True)
node.setPropertyValue("confidence", True)
node.setPropertyValue("threshold", 75)
node.setPropertyValue("improve_accuracy", 3)
node.setPropertyValue("inc_user_measure", True)
# "Define User Measure..."
node.setPropertyValue("user_if", "@TARGET = @PREDICTED")
node.setPropertyValue("user_then", "101")
node.setPropertyValue("user_else", "1")
node.setPropertyValue("user_compute", ["Mean", "Sum"])
node.setPropertyValue("by_fields", ["Drug"])
# "Output" tab
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/output/analysis_out.html")
```

Table 1. analysisnode properties

analysisnode properties	Data type	Property description
<code>output_mode</code>	<code>Screen File</code>	Used to specify target location for output generated from the output node.
<code>use_output_name</code>	<code>flag</code>	Specifies whether a custom output name is used.
<code>output_name</code>	<code>string</code>	If <code>use_output_name</code> is true, specifies the name to use.
<code>output_format</code>	<code>Text (.txt) HTML (.html)</code> <code>Output (.cou)</code>	Used to specify the type of output.
<code>by_fields</code>	<code>list</code>	
<code>full_filename</code>	<code>string</code>	If disk, data, or HTML output, the name of the output file.
<code>coincidence</code>	<code>flag</code>	
<code>performance</code>	<code>flag</code>	
<code>evaluation_binary</code>	<code>flag</code>	
<code>confidence</code>	<code>flag</code>	
<code>threshold</code>	<code>number</code>	
<code>improve_accuracy</code>	<code>number</code>	
<code>field_detection_meth</code>	<code>Metadata Name</code>	Determines how predicted fields are matched to the original target field. Specify <code>Metadata</code> or <code>Name</code> .
<code>inc_user_measure</code>	<code>flag</code>	
<code>user_if</code>	<code>expr</code>	
<code>user_then</code>	<code>expr</code>	
<code>user_else</code>	<code>expr</code>	
<code>user_compute</code>	<code>[Mean Sum Min Max SDev]</code>	

dataauditnode properties



The Data Audit node provides a comprehensive first look at the data, including summary statistics, histograms and distribution for each field, as well as information on outliers, missing values, and extremes. Results are displayed in an easy-to-read matrix that can be sorted and used to generate full-size graphs and data preparation nodes.

Example

```
filenode = stream.createAt("variablefile", "File", 100, 100)
filenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
```

```

node = stream.createAt("dataaudit", "My node", 196, 100)
stream.link(filenode, node)
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("fields", ["Age", "Na", "K"])
node.setPropertyValue("display_graphs", True)
node.setPropertyValue("basic_stats", True)
node.setPropertyValue("advanced_stats", True)
node.setPropertyValue("median_stats", False)
node.setPropertyValue("calculate", ["Count", "Breakdown"])
node.setPropertyValue("outlier_detection_method", "std")
node.setPropertyValue("outlier_detection_std_outlier", 1.0)
node.setPropertyValue("outlier_detection_std_extreme", 3.0)
node.setPropertyValue("output_mode", "Screen")

```

Table 1. dataauditnode properties

dataauditnode properties	Data type	Property description
custom_fields	flag	
fields	[field1 ... fieldN]	
overlay	field	
display_graphs	flag	Used to turn the display of graphs in the output matrix on or off.
basic_stats	flag	
advanced_stats	flag	
median_stats	flag	
calculate	Count Breakdown	Used to calculate missing values. Select either, both, or neither calculation method.
outlier_detection_method	std iqr	Used to specify the detection method for outliers and extreme values.
outlier_detection_std_outlier	number	If outlier_detection_method is std , specifies the number to use to define outliers.
outlier_detection_std_extreme	number	If outlier_detection_method is std , specifies the number to use to define extreme values.
outlier_detection_iqr_outlier	number	If outlier_detection_method is iqr , specifies the number to use to define outliers.
outlier_detection_iqr_extreme	number	If outlier_detection_method is iqr , specifies the number to use to define extreme values.
use_output_name	flag	Specifies whether a custom output name is used.
output_name	string	If use_output_name is true, specifies the name to use.
output_mode	Screen File	Used to specify target location for output generated from the output node.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Used to specify the type of output.
paginate_output	flag	When the output_format is HTML , causes the output to be separated into pages.
lines_per_page	number	When used with paginate_output , specifies the lines per page of output.
full_filename	string	

extensionoutputnode properties



The Extension Output node enables you to analyze data and the results of model scoring using your own custom R or Python for Spark script. The output of the analysis can be text or graphical. The output is added to the Output tab of the manager pane; alternatively, the output can be redirected to a file.

Python for Spark example

```

##### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_output", "extension_output")
node.setPropertyValue("syntax_type", "Python")

python_script = """
import json

```

```

import spss.pyspark.runtime

ctx = spss.pyspark.runtime.getContext()
df = ctx.getSparkInputData()
schema = df.dtypes[:]
print df
"""

node.setPropertyValue("python_syntax", python_script)

```

R example

```

##### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_syntax", "print(modelerData$Age)")

```

Table 1. extensionoutputnode properties

extensionoutputnode properties	Data type	Property description
syntax_type	<i>R Python</i>	Specify which script runs – R or Python (R is the default).
r_syntax	<i>string</i>	R scripting syntax for model scoring.
python_syntax	<i>string</i>	Python scripting syntax for model scoring.
convert_flags	<i>StringsAndDouble s LogicalValues</i>	Option to convert flag fields.
convert_missing	<i>flag</i>	Option to convert missing values to the R NA value.
convert_datetime	<i>flag</i>	Option to convert variables with date or datetime formats to R date/time formats.
convert_datetime_class	<i>POSIXct POSIXlt</i>	Options to specify to what format variables with date or datetime formats are converted.
output_to	<i>Screen File</i>	Specify the output type (Screen or File).
output_type	<i>Graph Text</i>	Specify whether to produce graphical or text output.
full_filename	<i>string</i>	File name to use for the generated output.
graph_file_type	<i>HTML COU</i>	File type for the output file (.html or .cou).
text_file_type	<i>HTML TEXT COU</i>	Specify the file type for text output (.html, .txt, or .cou).

kdeexport properties



Kernel Density Estimation (KDE)© uses the Ball Tree or KD Tree algorithms for efficient queries, and combines concepts from unsupervised learning, feature engineering, and data modeling. Neighbor-based approaches such as KDE are some of the most popular and useful density estimation techniques. The KDE Modeling and KDE Simulation nodes in SPSS® Modeler expose the core features and commonly used parameters of the KDE library. The nodes are implemented in Python.

Table 1. kdeexport properties

kdeexport properties	Data type	Property description
bandwidth	<i>double</i>	Default is 1.
kernel	<i>string</i>	The kernel to use: gaussian or tophat . Default is gaussian .
algorithm	<i>string</i>	The tree algorithm to use: kd_tree , ball_tree , or auto . Default is auto .
metric	<i>string</i>	The metric to use when calculating distance. For the kd_tree algorithm, choose from: Euclidean , Chebyshev , Cityblock , Minkowski , Manhattan , Infinity , P , L2 , or L1 . For the ball_tree algorithm, choose from: Euclidian , Braycurtis , Chebyshev , Canberra , Cityblock , Dice , Hamming , Infinity , Jaccard , L1 , L2 , Minkowski , Matching , Manhattan , P , Rogersanimoto , Russellrao , Sokalmichener , Sokalsneath , or Kulsinski . Default is Euclidean .
atol	<i>float</i>	The desired absolute tolerance of the result. A larger tolerance will generally lead to faster execution. Default is 0 . 0.
rtol	<i>float</i>	The desired relative tolerance of the result. A larger tolerance will generally lead to faster execution. Default is 1E-8.
breadthFirst	<i>boolean</i>	Set to True to use a breadth-first approach. Set to False to use a depth-first approach. Default is True .
LeafSize	<i>integer</i>	The leaf size of the underlying tree. Default is 40. Changing this value may significantly impact the performance.
pValue	<i>double</i>	Specify the P Value to use if you're using Minkowski for the metric. Default is 1 . 5.

matrixnode properties



The Matrix node creates a table that shows relationships between fields. It is most commonly used to show the relationship between two symbolic fields, but it can also show relationships between flag fields or numeric fields.

Example

```
node = stream.create("matrix", "My node")
# "Settings" tab
node.setPropertyValue("fields", "Numerics")
node.setPropertyValue("row", "K")
node.setPropertyValue("column", "Na")
node.setPropertyValue("cell_contents", "Function")
node.setPropertyValue("function_field", "Age")
node.setPropertyValue("function", "Sum")
# "Appearance" tab
node.setPropertyValue("sort_mode", "Ascending")
node.setPropertyValue("highlight_top", 1)
node.setPropertyValue("highlight_bottom", 5)
node.setPropertyValue("display", ["Counts", "Expected", "Residuals"])
node.setPropertyValue("include_totals", True)
# "Output" tab
node.setPropertyValue("full_filename", "C:/output/matrix_output.html")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("paginate_output", True)
node.setPropertyValue("lines_per_page", 50)
```

Table 1. matrixnode properties

matrixnode properties	Data type	Property description
fields	Selected Flags Numerics	
row	field	
column	field	
include_missing_values	flag	Specifies whether user-missing (blank) and system missing (null) values are included in the row and column output.
cell_contents	CrossTabs Function	
function_file	string	
function	Sum Mean Min Max SDev	
sort_mode	Unsorted Ascending Descending	
highlight_top	number	If non-zero, then true.
highlight_bottom	number	If non-zero, then true.
display	[Counts Expected Residuals RowPct ColumnPct TotalPct]	
include_totals	flag	
use_output_name	flag	Specifies whether a custom output name is used.
output_name	string	If <code>use_output_name</code> is true, specifies the name to use.
output_mode	Screen File	Used to specify target location for output generated from the output node.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Used to specify the type of output. Both the <code>Formatted</code> and <code>Delimited</code> formats can take the modifier <code>transposed</code> , which transposes the rows and columns in the table.
paginate_output	flag	When the <code>output_format</code> is <code>HTML</code> , causes the output to be separated into pages.
lines_per_page	number	When used with <code>paginate_output</code> , specifies the lines per page of output.
full_filename	string	

meansnode properties



The Means node compares the means between independent groups or between pairs of related fields to test whether a significant difference exists. For example, you could compare mean revenues before and after running a promotion or



compare revenues from customers who did not receive the promotion with those who did.

Example

```
node = stream.create("means", "My node")
node.setPropertyValue("means_mode", "BetweenFields")
node.setPropertyValue("paired_fields", [["OPEN_BAL", "CURR_BAL"]])
node.setPropertyValue("label_correlations", True)
node.setPropertyValue("output_view", "Advanced")
node.setPropertyValue("output_mode", "File")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/output/means_output.html")
```

Table 1. meansnode properties

meansnode properties	Data type	Property description
means_mode	BetweenGroups BetweenFields	Specifies the type of means statistic to be executed on the data.
test_fields	[field1 ... fieldn]	Specifies the test field when means_mode is set to BetweenGroups .
grouping_field	field	Specifies the grouping field.
paired_fields	[[field1 field2] [field3 field4] ...]	Specifies the field pairs to use when means_mode is set to BetweenFields .
label_correlations	flag	Specifies whether correlation labels are shown in output. This setting applies only when means_mode is set to BetweenFields .
correlation_mode	Probability Absolute	Specifies whether to label correlations by probability or absolute value.
weak_label	string	
medium_label	string	
strong_label	string	
weak_below_probability	number	When correlation_mode is set to Probability , specifies the cutoff value for weak correlations. This must be a value between 0 and 1—for example, 0.90.
strong_above_probability	number	Cutoff value for strong correlations.
weak_below_absolute	number	When correlation_mode is set to Absolute , specifies the cutoff value for weak correlations. This must be a value between 0 and 1—for example, 0.90.
strong_above_absolute	number	Cutoff value for strong correlations.
unimportant_label	string	
marginal_label	string	
important_label	string	
unimportant_below	number	Cutoff value for low field importance. This must be a value between 0 and 1—for example, 0.90.
important_above	number	
use_output_name	flag	Specifies whether a custom output name is used.
output_name	string	Name to use.
output_mode	Screen File	Specifies the target location for output generated from the output node.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Specifies the type of output.
full_filename	string	
output_view	Simple Advanced	Specifies whether the simple or advanced view is displayed in the output.

reportnode properties



The Report node creates formatted reports containing fixed text as well as data and other expressions derived from the data. You specify the format of the report using text templates to define the fixed text and data output constructions. You can provide custom text formatting by using HTML tags in the template and by setting options on the Output tab. You can include data values and other conditional output by using CLEM expressions in the template.

Example

```

node = stream.create("report", "My node")
node.setPropertyValue("output format", "HTML")
node.setPropertyValue("full_filename", "C:/report_output.html")
node.setPropertyValue("lines_per_page", 50)
node.setPropertyValue("title", "Report node created by a script")
node.setPropertyValue("highlights", False)

```

Table 1. reportnode properties

reportnode properties	Data type	Property description
output_mode	Screen File	Used to specify target location for output generated from the output node.
output_format	HTML (.html) Text (.txt) Output (.cou)	Used to specify the type of file output.
format	Auto Custom	Used to choose whether output is automatically formatted or formatted using HTML included in the template. To use HTML formatting in the template, specify Custom.
use_output_name	flag	Specifies whether a custom output name is used.
output_name	string	If use_output_name is true, specifies the name to use.
text	string	
full_filename	string	
highlights	flag	
title	string	
lines_per_page	number	

setglobalsnode properties



The Set Globals node scans the data and computes summary values that can be used in CLEM expressions. For example, you can use this node to compute statistics for a field called *age* and then use the overall mean of *age* in CLEM expressions by inserting the function @GLOBAL_MEAN(*age*).

Example

```

node = stream.create("setglobals", "My node")
node.setKeyedPropertyValue("globals", "Na", ["Max", "Sum", "Mean"])
node.setKeyedPropertyValue("globals", "K", ["Max", "Sum", "Mean"])
node.setKeyedPropertyValue("globals", "Age", ["Max", "Sum", "Mean", "SDev"])
node.setPropertyValue("clear_first", False)
node.setPropertyValue("show_preview", True)

```

Table 1. setglobalsnode properties

setglobalsnode properties	Data type	Property description
globals	[Sum Mean Min Max SDev]	Structured property where fields to be set must be referenced with the following syntax: node.setKeyedPropertyValue("globals", "Age", ["Max", "Sum", "Mean", "SDev"])
clear_first	flag	
show_preview	flag	

simevalnode properties



The Simulation Evaluation node evaluates a specified predicted target field, and presents distribution and correlation information about the target field.

Table 1. simevalnode properties

simevalnode properties	Data type	Property description
target	field	
iteration	field	

simevalnode properties	Data type	Property description
<code>presorted_by_iteration</code>	<code>boolean</code>	
<code>max_iterations</code>	<code>number</code>	
<code>tornado_fields</code>	<code>[field1...fieldN]</code>	
<code>plot_pdf</code>	<code>boolean</code>	
<code>plot_cdf</code>	<code>boolean</code>	
<code>show_ref_mean</code>	<code>boolean</code>	
<code>show_ref_median</code>	<code>boolean</code>	
<code>show_ref_sigma</code>	<code>boolean</code>	
<code>num_ref_sigma</code>	<code>number</code>	
<code>show_ref_pct</code>	<code>boolean</code>	
<code>ref_pct_bottom</code>	<code>number</code>	
<code>ref_pct_top</code>	<code>number</code>	
<code>show_ref_custom</code>	<code>boolean</code>	
<code>ref_custom_values</code>	<code>[number1...numberN]</code>	
<code>category_values</code>	<code>Category Probabilities Both</code>	
<code>category_groups</code>	<code>Categories Iterations</code>	
<code>create_pct_table</code>	<code>boolean</code>	
<code>pct_table</code>	<code>Quartiles Intervals Custom</code>	
<code>pct_intervals_num</code>	<code>number</code>	
<code>pct_custom_values</code>	<code>[number1...numberN]</code>	

simfitnode properties



The Simulation Fitting node examines the statistical distribution of the data in each field and generates (or updates) a Simulation Generate node, with the best fitting distribution assigned to each field. The Simulation Generate node can then be used to generate simulated data.

Table 1. simfitnode properties

simfitnode properties	Data type	Property description
<code>build</code>	<code>Node XMLEXport Both</code>	
<code>use_source_node_name</code>	<code>boolean</code>	
<code>source_node_name</code>	<code>string</code>	The custom name of the source node that is either being generated or updated.
<code>use_cases</code>	<code>All LimitFirstN</code>	
<code>use_case_limit</code>	<code>integer</code>	
<code>fit_criterion</code>	<code>AndersonDarling KolmogorovSmirnov</code>	
<code>num_bins</code>	<code>integer</code>	
<code>parameter_xml_filename</code>	<code>string</code>	
<code>generate_parameter_import</code>	<code>boolean</code>	

statisticsnode properties



The Statistics node provides basic summary information about numeric fields. It calculates summary statistics for individual fields and correlations between fields.

Example

```
node = stream.create("statistics", "My node")
# "Settings" tab
node.setPropertyValue("examine", ["Age", "BP", "Drug"])
node.setPropertyValue("statistics", ["mean", "sum", "sdev"])
node.setPropertyValue("correlate", ["BP", "Drug"])
```

```

# "Correlation Labels..." section
node.setPropertyValue("label_correlations", True)
node.setPropertyValue("weak_below_absolute", 0.25)
node.setPropertyValue("weak_label", "lower quartile")
node.setPropertyValue("strong_above_absolute", 0.75)
node.setPropertyValue("medium_label", "middle quartiles")
node.setPropertyValue("strong_label", "upper quartile")
# "Output" tab
node.setPropertyValue("full_filename", "c:/output/statistics_output.html")
node.setPropertyValue("output_format", "HTML")

```

Table 1. statisticsnode properties

statisticsnode properties	Data type	Property description
use_output_name	flag	Specifies whether a custom output name is used.
output_name	string	If use_output_name is true, specifies the name to use.
output_mode	Screen File	Used to specify target location for output generated from the output node.
output_format	Text (.txt) HTML (.html) Output (.cou)	Used to specify the type of output.
full_filename	string	
examine	list	
correlate	list	
statistics	[count mean sum min max range variance sdev semean median mode]	
correlation_mode	Probability Absolute	Specifies whether to label correlations by probability or absolute value.
label_correlations	flag	
weak_label	string	
medium_label	string	
strong_label	string	
weak_below_probability	number	When correlation_mode is set to Probability , specifies the cutoff value for weak correlations. This must be a value between 0 and 1—for example, 0.90.
strong_above_probability	number	Cutoff value for strong correlations.
weak_below_absolute	number	When correlation_mode is set to Absolute , specifies the cutoff value for weak correlations. This must be a value between 0 and 1—for example, 0.90.
strong_above_absolute	number	Cutoff value for strong correlations.

statisticsoutputnode Properties



The Statistics Output node allows you to call an IBM® SPSS® Statistics procedure to analyze your IBM SPSS Modeler data. A wide variety of IBM SPSS Statistics analytical procedures is available. This node requires a licensed copy of IBM SPSS Statistics.

The properties for this node are described under [statisticsoutputnode Properties](#).

tablenode properties



The Table node displays the data in table format, which can also be written to a file. This is useful anytime that you need to inspect your data values or export them in an easily readable form.

Example

```

node = stream.create("table", "My node")
node.setPropertyValue("highlight_expr", "Age > 30")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("transpose_data", True)
node.setPropertyValue("full_filename", "C:/output/table_output.htm")
node.setPropertyValue("paginate_output", True)
node.setPropertyValue("lines_per_page", 50)

```

Table 1. tablenode properties

tablenode properties	Data type	Property description
<code>full_filename</code>	<code>string</code>	If disk, data, or HTML output, the name of the output file.
<code>use_output_name</code>	<code>flag</code>	Specifies whether a custom output name is used.
<code>output_name</code>	<code>string</code>	If <code>use_output_name</code> is true, specifies the name to use.
<code>output_mode</code>	<code>Screen File</code>	Used to specify target location for output generated from the output node.
<code>output_format</code>	<code>Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)</code>	Used to specify the type of output.
<code>transpose_data</code>	<code>flag</code>	Transposes the data before export so that rows represent fields and columns represent records.
<code>paginate_output</code>	<code>flag</code>	When the <code>output_format</code> is <code>HTML</code> , causes the output to be separated into pages.
<code>lines_per_page</code>	<code>number</code>	When used with <code>paginate_output</code> , specifies the lines per page of output.
<code>highlight_expr</code>	<code>string</code>	
<code>output</code>	<code>string</code>	A read-only property that holds a reference to the last table built by the node.
<code>value_labels</code>	<code>[[Value LabelString] [Value LabelString] ...]</code>	Used to specify labels for value pairs.
<code>display_places</code>	<code>integer</code>	Sets the number of decimal places for the field when displayed (applies only to fields with REAL storage). A value of -1 will use the stream default.
<code>export_places</code>	<code>integer</code>	Sets the number of decimal places for the field when exported (applies only to fields with REAL storage). A value of -1 will use the stream default.
<code>decimal_separator</code>	<code>DEFAULT PERIOD COMMA</code>	Sets the decimal separator for the field (applies only to fields with REAL storage).
<code>date_format</code>	<pre>"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.XX" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY Q YYYY WW WK YYYY</pre>	Sets the date format for the field (applies only to fields with <code>DATE</code> or <code>TIMESTAMP</code> storage).
<code>time_format</code>	<pre>"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" " (H) H: (M) M: (S) S" "(H) H: (M) M" "(M) M: (S) S" "HH.MM.SS" "HH.MM" "MM.SS" "(H) H. (M) M. (S) S" "(H) H. (M) M" "(M) M. (S) S"</pre>	Sets the time format for the field (applies only to fields with <code>TIME</code> or <code>TIMESTAMP</code> storage).
<code>column_width</code>	<code>integer</code>	Sets the column width for the field. A value of -1 will set column width to <code>Auto</code> .
<code>justify</code>	<code>AUTO CENTER LEFT RIGHT</code>	Sets the column justification for the field.

transformnode properties

The Transform node allows you to select and visually preview the results of transformations before applying them to selected



fields.

Example

```
node = stream.create("transform", "My node")
node.setPropertyValue("fields", ["AGE", "INCOME"])
node.setPropertyValue("formula", "Select")
node.setPropertyValue("formula_log_n", True)
node.setPropertyValue("formula_log_n_offset", 1)
```

Table 1. transformnode properties

transformnode properties	Data type	Property description
fields	[field1... fieldn]	The fields to be used in the transformation.
formula	All Select	Indicates whether all or selected transformations should be calculated.
formula_inverse	flag	Indicates if the inverse transformation should be used.
formula_inverse_offset	number	Indicates a data offset to be used for the formula. Set as 0 by default, unless specified by user.
formula_log_n	flag	Indicates if the \log_n transformation should be used.
formula_log_n_offset	number	
formula_log_10	flag	Indicates if the \log_{10} transformation should be used.
formula_log_10_offset	number	
formula_exponential	flag	Indicates if the exponential transformation (e^x) should be used.
formula_square_root	flag	Indicates if the square root transformation should be used.
use_output_name	flag	Specifies whether a custom output name is used.
output_name	string	If use_output_name is true, specifies the name to use.
output_mode	Screen File	Used to specify target location for output generated from the output node.
output_format	HTML (.html) Output (.cou)	Used to specify the type of output.
paginate_output	flag	When the output_format is HTML, causes the output to be separated into pages.
lines_per_page	number	When used with paginate_output , specifies the lines per page of output.
full_filename	string	Indicates the file name to be used for the file output.

Export Node Properties

- [Common Export Node Properties](#)
- [asexport Properties](#)
- [cognoselexportnode Properties](#)
- [databaseexportnode properties](#)
- [datacollectionexportnode Properties](#)
- [excelexportnode Properties](#)
- [extensionexportnode_properties](#)
- [jsonexportnode Properties](#)
- [outputfilenode Properties](#)
- [publishernode Properties](#)
- [sasexportnode Properties](#)
- [statisticsexportnode Properties](#)
- [tm1odataexport Node Properties](#)
- [tm1export Node Properties \(deprecated\)](#)
- [xmlexportnode Properties](#)

Common Export Node Properties

The following properties are common to all export nodes.

Table 1. Common export node properties

Property	Values	Property description
publish_path	string	Enter the rootname name to be used for the published image and parameter files.

Property	Values	Property description
<code>publish_metadata</code>	<code>flag</code>	Specifies if a metadata file is produced that describes the inputs and outputs of the image and their data models.
<code>publish_use_parameters</code>	<code>flag</code>	Specifies if stream parameters are included in the *.par file.
<code>publish_parameters</code>	<code>string list</code>	Specify the parameters to be included.
<code>execute_mode</code>	<code>export_data publish</code>	Specifies whether the node executes without publishing the stream, or if the stream is automatically published when the node is executed.

aselexport Properties

The Analytic Server export enables you to run a stream on Hadoop Distributed File System (HDFS).

Example

```
node.setPropertyValue("use_default_as", False)
node.setPropertyValue("connection",
["false","9.119.141.141","9080","analyticserver","ibm","admin","admin","false","","","",""])

```

Table 1. asexport properties

aselexport properties	Data type	Property description
<code>data_source</code>	<code>string</code>	The name of the data source.
<code>export_mode</code>	<code>string</code>	Specifies whether to append exported data to the existing data source, or to overwrite the existing data source.
<code>use_default_as</code>	<code>boolean</code>	If set to <code>True</code> , uses the default Analytic Server connection configured in the server options.cfg file. If set to <code>False</code> , uses the connection of this node.
<code>connection</code>	<code>["string", "string", "string", "string", "string", "string", "string", "string", "string", "string", "string", "string"]</code>	A list property containing the Analytic Server connection details. The format is: <code>["is_secure_connect", "server_url", "server_port", "context_root", "consumer", "user_name", "password", "use-kerberos-auth", "kerberos-krb5-config-file-path", "kerberos-jaas-config-file-path", "kerberos-krb5-service-principal-name", "enable-kerberos-debug"]</code> Where: <code>is_secure_connect</code> : indicates whether secure connection is used, and is either <code>true</code> or <code>false</code> . <code>use-kerberos-auth</code> : indicates whether kerberos authentication is used, and is either <code>true</code> or <code>false</code> . <code>enable-kerberos-debug</code> : indicates whether the debug mode of kerberos authentication is used, and is either <code>true</code> or <code>false</code> .

cognosexportnode Properties



The IBM Cognos Export node exports data in a format that can be read by Cognos databases.

For this node, you must define a Cognos connection and an ODBC connection.

Cognos connection

The properties for the Cognos connection are as follows.

Table 1. cognosexportnode properties

cognosexport node properties	Data type	Property description

cognosexport node properties	Data type	Property description
cognos_connection	["string","flag","string","string"]	A list property containing the connection details for the Cognos server. The format is: ["Cognos_server_URL", "login_mode", "namespace", "username", "password"] where: Cognos_server_URL is the URL of the Cognos server containing the source. login_mode indicates whether anonymous login is used, and is either true or false; if set to true, the following fields should be set to "". namespace specifies the security authentication provider used to log on to the server. username and password are those used to log on to the Cognos server. Instead of login_mode, the following modes are also available: <ul style="list-style-type: none"> anonymousMode. For example: ['Cognos_server_url', 'anonymousMode', 'namespace', 'username', 'password'] credentialMode. For example: ['Cognos_server_url', 'credentialMode', 'namespace', 'username', 'password']
		<ul style="list-style-type: none"> storedCredentialMode. For example: ['Cognos_server_url', 'storedCredentialMode', 'stored_credential_name'] Where stored_credential_name is the name of a Cognos credential in the repository.
cognos_package_name	string	The path and name of the Cognos package to which you are exporting data, for example: /Public Folders/MyPackage
cognos_datasource	string	
cognos_export_mode	Publish ExportFile	
cognos_filename	string	

ODBC connection

The properties for the ODBC connection are identical to those listed for **databaseexportnode** in the next section, with the exception that the **datasource** property is not valid.

databaseexportnode properties

	The Database export node writes data to an ODBC-compliant relational data source. In order to write to an ODBC data source, the data source must exist and you must have write permission for it.
---	---

Example

```
'''
Assumes a datasource named "MyDatasource" has been configured
'''

stream = modeler.script.stream()
db_exportnode = stream.createAt("databaseexport", "DB Export", 200, 200)
applynn = stream.findByName("applyneuralnetwork", None)
stream.link(applynn, db_exportnode)

# Export tab
db_exportnode.setPropertyValue("username", "user")
db_exportnode.setPropertyValue("datasource", "MyDatasource")
db_exportnode.setPropertyValue("password", "password")
db_exportnode.setPropertyValue("table_name", "predictions")
db_exportnode.setPropertyValue("write_mode", "Create")
db_exportnode.setPropertyValue("generate_import", True)
db_exportnode.setPropertyValue("drop_existing_table", True)
db_exportnode.setPropertyValue("delete_existing_rows", True)
db_exportnode.setPropertyValue("default_string_size", 32)

# Schema dialog
db_exportnode.setKeyedPropertyValue("type", "region", "VARCHAR(10)")
db_exportnode.setKeyedPropertyValue("export_db_primarykey", "id", True)
db_exportnode.setPropertyValue("use_custom_create_table_command", True)
db_exportnode.setPropertyValue("custom_create_table_command", "My SQL Code")

# Indexes dialog
db_exportnode.setPropertyValue("use_custom_create_index_command", True)
db_exportnode.setPropertyValue("custom_create_index_command", "CREATE BITMAP INDEX <index-name>
```

```

ON <table-name> <(index-columns)>"')
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", ["fields", ["id", "region"]])

```

Table 1. databaseexportnode properties

databaseexportnode properties	Data type	Property description
datasource	<i>string</i>	
username	<i>string</i>	
password	<i>string</i>	
epassword	<i>string</i>	This slot is read-only during execution. To generate an encoded password, use the Password Tool available from the Tools menu. See the topic Generating an encoded password for more information.
table_name	<i>string</i>	
write_mode	<i>Create</i> <i>Append Merge</i>	
map	<i>string</i>	Maps a stream field name to a database column name (valid only if write_mode is Merge). For a merge, all fields must be mapped in order to be exported. Field names that do not exist in the database are added as new columns.
key_fields	<i>list</i>	Specifies the stream field that is used for key; map property shows what this corresponds to in the database.
join	<i>Database Add</i>	
drop_existing_table	<i>flag</i>	
delete_existing_rows	<i>flag</i>	
default_string_size	<i>integer</i>	
type		Structured property used to set the schema type.
generate_import	<i>flag</i>	
use_custom_create_table_command	<i>flag</i>	Use the <i>custom_create_table</i> slot to modify the standard CREATE TABLE SQL command.
custom_create_table_command	<i>string</i>	Specifies a string command to use in place of the standard CREATE TABLE SQL command.
use_batch	<i>flag</i>	The following properties are advanced options for database bulk-loading. A True value for Use_batch turns off row-by-row commits to the database.
batch_size	<i>number</i>	Specifies the number of records to send to the database before committing to memory.
bulk_loading	<i>Off</i> <i>ODBC</i> <i>External</i>	Specifies the type of bulk-loading. Additional options for ODBC and External are listed below.
not_logged	<i>flag</i>	
odbc_binding	<i>Row Column</i>	Specify row-wise or column-wise binding for bulk-loading via ODBC.
loader_delimit_mode	<i>Tab Space</i> <i>Other</i>	For bulk-loading via an external program, specify type of delimiter. Select Other in conjunction with the loader_other_delimiter property to specify delimiters, such as the comma (,).
loader_other_delimiter	<i>string</i>	
specify_data_file	<i>flag</i>	A True flag activates the data_file property below, where you can specify the filename and path to write to when bulk-loading to the database.
data_file	<i>string</i>	
specify_loader_program	<i>flag</i>	A True flag activates the loader_program property below, where you can specify the name and location of an external loader script or program.
loader_program	<i>string</i>	
gen_logfile	<i>flag</i>	A True flag activates the logfile_name below, where you can specify the name of a file on the server to generate an error log.
logfile_name	<i>string</i>	
check_table_size	<i>flag</i>	A True flag allows table checking to ensure that the increase in database table size corresponds to the number of rows exported from IBM® SPSS® Modeler.
loader_options	<i>string</i>	Specify additional arguments, such as -comment and -specialdir , to the loader program.
export_db_primarykey	<i>flag</i>	Specifies whether a given field is a primary key.
use_custom_create_index_command	<i>flag</i>	If true , enables custom SQL for all indexes.
custom_create_index_command	<i>string</i>	Specifies the SQL command used to create indexes when custom SQL is enabled. (This value can be overridden for specific indexes as indicated below.)
indexes.INDEXNAME.fields		Creates the specified index if necessary and lists field names to be included in that index.
INDEXNAME "use_custom_create_index_command"	<i>flag</i>	Used to enable or disable custom SQL for a specific index. See examples after the following table.

databaseexportnode properties	Data type	Property description
<code>INDEXNAME "custom_create_index_command"</code>	<code>string</code>	Specifies the custom SQL used for the specified index. See examples after the following table.
<code>indexes.INDEXNAME.rmove</code>	<code>flag</code>	If <code>True</code> , removes the specified index from the set of indexes.
<code>table_space</code>	<code>string</code>	Specifies the table space that will be created.
<code>use_partition</code>	<code>flag</code>	Specifies that the distribute hash field will be used.
<code>partition_field</code>	<code>string</code>	Specifies the contents of the distribute hash field.

Note: For some databases, you can specify that database tables are created for export with compression (for example, the equivalent of `CREATE TABLE MYTABLE (...) COMPRESS YES` in SQL). The properties `use_compression` and `compression_mode` are provided to support this feature, as follows.

Table 2. databaseexportnode properties using compression features

databaseexportnode properties	Data type	Property description
<code>use_compression</code>	<code>Boolean</code>	If set to <code>True</code> , creates tables for export with compression.
<code>compression_mode</code>	<code>Row Page</code> <code>Default Direct_Load_Operations</code> <code>All_Operations Basic OLTP Query_High</code> <code>Query_Low Archive_High Archive_Low</code>	Sets the level of compression for SQL Server databases. Sets the level of compression for Oracle databases. Note that the values <code>OLTP</code> , <code>Query_High</code> , <code>Query_Low</code> , <code>Archive_High</code> , and <code>Archive_Low</code> require a minimum of Oracle 11gR2.

Example showing how to change the `CREATE INDEX` command for a specific index:

```
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", ["use_custom_create_index_command", True]) db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", ["custom_create_index_command", "CREATE BITMAP INDEX <index-name> ON <table-name> <(index-columns)>"])
```

Alternatively, this can be done via a hash table:

```
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", {"fields": ["id", "region"], "use_custom_create_index_command": True, "custom_create_index_command": "CREATE INDEX <index-name> ON <table-name> <(index-columns)>"})
```

datacollectionexportnode Properties

	The Data Collection export node outputs data in the format used by Data Collection market research software. A Data Collection Data Library must be installed to use this node.
---	---

Example

```
stream = modeler.script.stream()
datacollectionexportnode = stream.createAt("datacollectionexport", "Data Collection", 200, 200)
datacollectionexportnode.setPropertyValue("metadata_file", "c:\\\\museums.mdd")
datacollectionexportnode.setPropertyValue("merge_metadata", "Overwrite")
datacollectionexportnode.setPropertyValue("casedata_file", "c:\\\\museumdata.sav")
datacollectionexportnode.setPropertyValue("generate_import", True)
datacollectionexportnode.setPropertyValue("enable_system_variables", True)
```

Table 1. datacollectionexportnode properties

datacollectionexportnode properties	Data type	Property description
<code>metadata_file</code>	<code>string</code>	The name of the metadata file to export.
<code>merge_metadata</code>	<code>Overwrite</code> <code>MergeCurrent</code>	
<code>enable_system_variables</code>	<code>flag</code>	Specifies whether the exported <code>.mdd</code> file should include Data Collection system variables.
<code>casedata_file</code>	<code>string</code>	The name of the <code>.sav</code> file to which case data is exported.
<code>generate_import</code>	<code>flag</code>	

Related information

- [Node properties overview](#)
- [featureselectionnode properties](#)
- [Output node properties](#)
- [analysisnode properties](#)

- [dataauditnode properties](#)
- [matrixnode properties](#)
- [meansnode properties](#)
- [reportnode properties](#)
- [setglobalsnode properties](#)
- [statisticsnode properties](#)
- [tablenode.properties](#)
- [transformnode properties](#)
- [databaseexportnode properties](#)
- [excelexportnode Properties](#)
- [outputfilenode Properties](#)
- [publishernode Properties](#)
- [sasexportnode Properties](#)
- [xmlexportnode Properties](#)
- [statisticsoutputnode Properties](#)
- [statisticsexportnode Properties](#)

excelexportnode Properties



The Excel export node outputs data in the Microsoft Excel .xlsx file format. Optionally, you can choose to launch Excel automatically and open the exported file when the node is executed.

Example

```
stream = modeler.script.stream()
excelexportnode = stream.createAt("excelexport", "Excel", 200, 200)
excelexportnode.setPropertyValue("full_filename", "C:/output/myexport.xlsx")
excelexportnode.setPropertyValue("excel_file_type", "Excel2007")
excelexportnode.setPropertyValue("inc_field_names", True)
excelexportnode.setPropertyValue("inc_labels_as_cell_notes", False)
excelexportnode.setPropertyValue("launch_application", True)
excelexportnode.setPropertyValue("generate_import", True)
```

Table 1. excelexportnode properties

excelexportnode properties	Data type	Property description
full_filename	string	
excel_file_type	Excel2007	
export_mode	Create Append	
inc_field_names	flag	Specifies whether field names should be included in the first row of the worksheet.
start_cell	string	Specifies starting cell for export.
worksheet_name	string	Name of the worksheet to be written.
launch_application	flag	Specifies whether Excel should be invoked on the resulting file. Note that the path for launching Excel must be specified in the Helper Applications dialog box (Tools menu, Helper Applications).
generate_import	flag	Specifies whether an Excel Import node should be generated that will read the exported data file.

extensionexportnode properties



With the Extension Export node, you can run R or Python for Spark scripts to export data.

Python for Spark example

```
#### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_export", "extension_export")
node.setPropertyValue("syntax_type", "Python")
```

```

python_script = """import spss.pyspark.runtime
from pyspark.sql import SQLContext
from pyspark.sql.types import *

ctx = spss.pyspark.runtime.getContext()
df = ctx.getSparkInputData()
print df.dtypes[:]
_newDF = df.select("Age", "Drug")
print _newDF.dtypes[:]

df.select("Age", "Drug").write.save("c:/data/ageAndDrug.json", format="json")
"""

node.setPropertyValue("python_syntax", python_script)

```

R example

```

#### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_syntax", """write.csv(modelerData, "C:/export.csv")""")

```

Table 1. extensionexportnode properties

extensionexportnode properties	Data type	Property description
syntax_type	<i>R Python</i>	Specify which script runs – R or Python (R is the default).
r_syntax	<i>string</i>	The R scripting syntax to run.
python_syntax	<i>string</i>	The Python scripting syntax to run.
convert_flags	<i>StringsAndDouble s LogicalValues</i>	Option to convert flag fields.
convert_missing	<i>flag</i>	Option to convert missing values to the R NA value.
convert_datetime	<i>flag</i>	Option to convert variables with date or datetime formats to R date/time formats.
convert_datetime_class	<i>POSIXct POSIXlt</i>	Options to specify to what format variables with date or datetime formats are converted.

jsonexportnode Properties



The JSON export node outputs data in JSON format. See [JSON Export node](#) for more information.

Table 1. jsonexportnode properties

jsonexportnode properties	Data type	Property description
full_filename	<i>string</i>	The complete filename, including path.
string_format	<i>records values</i>	Specify the format of the JSON string. Default is records .
generate_import	<i>flag</i>	Specifies whether a JSON Import node should be generated that will read the exported data file. Default is False .

outputfilenode Properties



The Flat File export node outputs data to a delimited text file. It is useful for exporting data that can be read by other analysis or spreadsheet software.

Example

```

stream = modeler.script.stream()
outputfile = stream.createAt("outputfile", "File Output", 200, 200)
outputfile.setPropertyValue("full_filename", "c:/output/flatfile_output.txt")
outputfile.setPropertyValue("write_mode", "Append")
outputfile.setPropertyValue("inc field names", False)
outputfile.setPropertyValue("use_newline_after_records", False)
outputfile.setPropertyValue("delimit_mode", "Tab")
outputfile.setPropertyValue("other_delimiter", ",")
outputfile.setPropertyValue("quote_mode", "Double")
outputfile.setPropertyValue("other_quote", "*")

```

```

outputfile.setPropertyValue("decimal_symbol", "Period")
outputfile.setPropertyValue("generate_import", True)

```

Table 1. outputfilenode properties

outputfilenode properties	Data type	Property description
full_filename	string	Name of output file.
write_mode	Overwrite Append	
inc_field_names	flag	
use_newline_after_records	flag	
delimit_mode	Comma Tab Space Other	
other_delimiter	char	
quote_mode	None Single Double Other	
other_quote	flag	
generate_import	flag	
encoding	StreamDefault SystemDefault "UTF-8"	

Related information

- [Node properties overview](#)
- [featureselectionnode properties](#)
- [Output node properties](#)
- [analysisnode properties](#)
- [dataauditnode properties](#)
- [matrixnode properties](#)
- [meansnode properties](#)
- [reportnode properties](#)
- [setglobalsnode properties](#)
- [statisticsnode properties](#)
- [tablenode properties](#)
- [transformnode properties](#)
- [databaseexportnode properties](#)
- [datacollectionexportnode Properties](#)
- [excelexportnode Properties](#)
- [publishernode Properties](#)
- [sasexportnode Properties](#)
- [xmlexportnode Properties](#)
- [statisticsoutputnode Properties](#)
- [statisticsexportnode Properties](#)

sasexportnode Properties



The SAS export node outputs data in SAS format, to be read into SAS or a SAS-compatible software package. Three SAS file formats are available: SAS for Windows/OS2, SAS for UNIX, or SAS Version 7/8.

Example

```

stream = modeler.script.stream()
sasexportnode = stream.createAt("sasexport", "SAS Export", 200, 200)
sasexportnode.setPropertyValue("full_filename", "c:/output/SAS_output.sas7bdat")
sasexportnode.setPropertyValue("format", "SAS8")
sasexportnode.setPropertyValue("export_names", "NamesAndLabels")
sasexportnode.setPropertyValue("generate_import", True)

```

Table 1. sasexportnode properties

sasexportnode properties	Data type	Property description
format	Windows UNIX SAS7 SAS8	Variant property label fields.
full_filename	string	
export_names	NamesAndLabels NamesAsLabels	Used to map field names from IBM® SPSS® Modeler upon export to IBM SPSS Statistics or SAS variable names.
generate_import	flag	

Related information

- [Node properties overview](#)
 - [featureselectionnode properties](#)
 - [Output node properties](#)
 - [analysisnode properties](#)
 - [dataauditnode properties](#)
 - [matrixnode properties](#)
 - [meansnode properties](#)
 - [reportnode properties](#)
 - [setglobalsnode properties](#)
 - [statisticsnode properties](#)
 - [tablenode properties](#)
 - [transformnode properties](#)
 - [databaseexportnode properties](#)
 - [datacollectionexportnode Properties](#)
 - [excelexportnode Properties](#)
 - [outputfilenode Properties](#)
 - [publishernode Properties](#)
 - [xmlexportnode Properties](#)
 - [statisticsoutputnode Properties](#)
 - [statisticsexportnode Properties](#)
-

staticsexportnode Properties



The Statistics Export node outputs data in IBM® SPSS® Statistics .sav or .zsav format. The .sav or .zsav files can be read by IBM SPSS Statistics Base and other products. This is also the format used for cache files in IBM SPSS Modeler.

The properties for this node are described under [statisticsexportnode Properties](#).

tm1odataexport Node Properties



The IBM Cognos TM1 Export node exports data in a format that can be read by Cognos TM1 databases.

Table 1. tm1odataexport node properties

tm1odataexpo rt node properties	Data type	Property description
<code>credential_t ype</code>	<code>inputCredential or storedCredentia l</code>	Used to indicate the credential type.
<code>input_creden tial</code>	<code>list</code>	When the <code>credential_type</code> is <code>inputCredential</code> ; specify the domain, user name and password.
<code>stored_crede ntial_name</code>	<code>string</code>	When the <code>credential_type</code> is <code>storedCredential</code> ; specify the name of credential on the C&DS server.
<code>selected_cub e</code>	<code>field</code>	The name of the cube to which you are exporting data. For example: <code>TM1_export.setPropertyValue("selected_cube", "plan_BudgetPlan")</code>

tm1odataexpo rt node properties	Data type	Property description
spss_field_t o_tm1_eleme nt_mapping	<i>list</i>	<p>The tm1 element to be mapped to must be part of the column dimension for selected cube view. The format is: [[[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...], [[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]]</p> <p>There are 2 lists to describe the mapping information. Mapping a leaf element to a dimension corresponds to example 2 below:</p> <p>Example 1: The first list: ([[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...]) is used for the TM1 Dimension map information.</p> <p>Each 3 value list indicates dimension mapping information. The third Boolean value is used to indicate if it selects an element of a dimension. For example: "[Field_1, Dimension_1, False]" means that Field_1 is mapped to Dimension_1; "[Element_1, Dimension_2, True]" means that Element_1 is selected for Dimension_2.</p> <p>Example 2: The second list: ([[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]) is used for the TM1 Measure Dimension Element map information.</p> <p>Each 3 value list indicates measure element mapping information. The third Boolean value is used to indicate the need to create a new element. "[Field_2, ExistMeasureElement, False]" means that Field_2 is mapped to the ExistMeasureElement; "[Field_3, NewMeasureElement, True]" means the NewMeasureElement needs to be the measure dimension chosen in selected_measure and that Field_3 is mapped to it.</p>
selected_me asure	<i>string</i>	<p>Specify the measure dimension.</p> <p>Example: <code>setPropertyValue("selected_measure", "Measures")</code></p>
connection_t ype	<i>AdminServer TM1Server</i>	Indicates the connection type. Default is AdminServer.
admin_host	<i>string</i>	The URL for the host name of the REST API. Required if the connection_type is AdminServer.
server_name	<i>string</i>	The name of the TM1 server selected from the admin_host. Required if the connection_type is AdminServer.
server_url	<i>string</i>	The URL for the TM1 Server REST API. Required if the connection_type is TM1Server.

tm1export Node Properties (deprecated)



The IBM Cognos TM1 Export node exports data in a format that can be read by Cognos TM1 databases.

Note: This node was deprecated in Modeler 18.0. The replacement node script name is *tm1odataexport*.

Table 1. tm1export node properties

tm1export node properties	Data type	Property description
pm_host	<i>string</i>	<p>Note: Only for version 16.0 and 17.0</p> <p>The host name. For example: <code>TM1_export.setPropertyValue("pm_host", "http://9.191.86.82:9510/pmhub/pm")</code></p>
tm1_connecti on	<i>["field","field", ... , "field"]</i>	<p>Note: Only for version 16.0 and 17.0</p> <p>A list property containing the connection details for the TM1 server. The format is: ["TM1_Server_Name", "tm1_username", "tm1_password"] For example: <code>TM1_export.setPropertyValue("tm1_connection", ["Planning Sample", "admin" "apple"])</code></p>
selected_cub e	<i>field</i>	<p>The name of the cube to which you are exporting data. For example:</p> <p><code>TM1_export.setPropertyValue("selected_cube", "plan_BudgetPlan")</code></p>

tm1export node properties	Data type	Property description
spssfield_tm1element_mapping	<i>list</i>	The tm1 element to be mapped to must be part of the column dimension for selected cube view. The format is: [[[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...], [[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]] There are 2 lists to describe the mapping information. Mapping a leaf element to a dimension corresponds to example 2 below: Example 1: The first list: ([[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...]) is used for the TM1 Dimension map information. Each 3 value list indicates dimension mapping information. The third Boolean value is used to indicate if it selects an element of a dimension. For example: "[Field_1, Dimension_1, False]" means that Field_1 is mapped to Dimension_1; "[Element_1, Dimension_2, True]" means that Element_1 is selected for Dimension_2. Example 2: The second list: ([[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]) is used for the TM1 Measure Dimension Element map information. Each 3 value list indicates measure element mapping information. The third Boolean value is used to indicate the need to create a new element. "[Field_2, ExistMeasureElement, False]" means that Field_2 is mapped to the ExistMeasureElement; "[Field_3, NewMeasureElement, True]" means the NewMeasureElement needs to be the measure dimension chosen in selected_measure and that Field_3 is mapped to it.
selected_measure	<i>string</i>	Specify the measure dimension. Example: setPropertyValue("selected_measure", "Measures")

xmlexportnode Properties

	The XML export node outputs data to a file in XML format. You can optionally create an XML source node to read the exported data back into the stream.
---	--

Example

```
stream = modeler.script.stream()
xmlexportnode = stream.createAt("xmlexport", "XML Export", 200, 200)
xmlexportnode.setPropertyValue("full_filename", "c:/export/data.xml")
xmlexportnode.setPropertyValue("map", ["/catalog/book/genre", "genre"], ["/catalog/book/title", "title"])
```

Table 1. xmlexportnode properties

xmlexportnode properties	Data type	Property description
full_filename	<i>string</i>	(required) Full path and file name of XML export file.
use_xml_schema	<i>flag</i>	Specifies whether to use an XML schema (XSD or DTD file) to control the structure of the exported data.
full_schema_filename	<i>string</i>	Full path and file name of XSD or DTD file to use. Required if use_xml_schema is set to true.
generate_import	<i>flag</i>	Generates an XML source node that will read the exported data file back into the stream.
records	<i>string</i>	XPath expression denoting the record boundary.
map	<i>string</i>	Maps field name to XML structure.

Related information

- [Node properties overview](#)
- [featureselectionnode properties](#)
- [Output node properties](#)
- [analysisnode properties](#)
- [dataauditnode properties](#)
- [matrixnode properties](#)
- [meansnode properties](#)
- [reportnode properties](#)
- [setglobalsnode properties](#)
- [statisticsnode properties](#)
- [tablenode properties](#)
- [transformnode properties](#)
- [databaseexportnode properties](#)
- [datacollectionexportnode Properties](#)
- [excelexportnode Properties](#)
- [outputfilenode Properties](#)
- [publishernode Properties](#)
- [sasexportnode Properties](#)

- [statisticsoutputnode Properties](#)
- [statisticsexportnode Properties](#)

IBM® SPSS® Statistics Node Properties

- [statisticsimportnode Properties](#)
- [statisticstransformnode properties](#)
- [statisticsmodelnode properties](#)
- [statisticsoutputnode Properties](#)
- [statisticsexportnode Properties](#)

statisticsimportnode Properties



The Statistics File node reads data from the .sav or .zsav file format used by IBM® SPSS® Statistics, as well as cache files saved in IBM SPSS Modeler, which also use the same format.

Example

```
stream = modeler.script.stream()
statisticsimportnode = stream.createAt("statisticsimport", "SAV Import", 200, 200)
statisticsimportnode.setPropertyValue("full_filename", "C:/data/drug1n.sav")
statisticsimportnode.setPropertyValue("import_names", True)
statisticsimportnode.setPropertyValue("import_data", True)
```

Table 1. statisticsimportnode properties

statisticsimportnode properties	Data type	Property description
full_filename	string	The complete filename, including path.
password	string	The password. The password parameter must be set before the file_encrypted parameter.
file_encrypted	flag	Whether or not the file is password protected.
import_names	NamesAndLabels LabelsAsNames	Method for handling variable names and labels.
import_data	DataAndLabels LabelsAsData	Method for handling values and labels.
use_field_format_for_storange	Boolean	Specifies whether to use IBM SPSS Statistics field format information when importing.

Related information

- [Node properties overview](#)
- [Source node common properties](#)
- [cognosimport Node Properties](#)
- [databasenode properties](#)
- [datacollectionimportnode Properties](#)
- [excelimportnode Properties](#)
- [evimportnode Properties](#)
- [fixedfilenode Properties](#)
- [sasimportnode Properties](#)
- [userinputnode properties](#)
- [variablefilenode Properties](#)
- [xmlimportnode Properties](#)

statisticstransformnode properties



The Statistics Transform node runs a selection of IBM® SPSS® Statistics syntax commands against data sources in IBM SPSS Modeler. This node requires a licensed copy of IBM SPSS Statistics.

Example

```

stream = modeler.script.stream()
statisticstransformnode = stream.createAt("statisticstransform", "Transform", 200, 200)
statisticstransformnode.setPropertyValue("syntax", "COMPUTE NewVar = Na + K.")
statisticstransformnode.setKeyedPropertyValue("new_name", "NewVar", "Mixed Drugs")
statisticstransformnode.setPropertyValue("check_before_saving", True)

```

Table 1. statisticstransformnode properties

statisticstransformnode properties	Data type	Property description
syntax	<i>string</i>	
check_before_saving	<i>flag</i>	Validates the entered syntax before saving the entries. Displays an error message if the syntax is invalid.
default_include	<i>flag</i>	See the topic filternode properties for more information.
include	<i>flag</i>	See the topic filternode properties for more information.
new_name	<i>string</i>	See the topic filternode properties for more information.

statisticsmodelnode properties

	The Statistics Model node enables you to analyze and work with your data by running IBM® SPSS® Statistics procedures that produce PMML. This node requires a licensed copy of IBM SPSS Statistics.
---	--

Example

```

stream = modeler.script.stream()
statisticsmodelnode = stream.createAt("statisticsmodel", "Model", 200, 200)
statisticsmodelnode.setPropertyValue("syntax", "COMPUTE NewVar = Na + K.")
statisticsmodelnode.setKeyedPropertyValue("new_name", "NewVar", "Mixed Drugs")

```

statisticsmodelnode properties	Data type	Property description
syntax	<i>string</i>	
default_include	<i>flag</i>	See the topic filternode properties for more information.
include	<i>flag</i>	See the topic filternode properties for more information.
new_name	<i>string</i>	See the topic filternode properties for more information.

statisticsoutputnode Properties

	The Statistics Output node allows you to call an IBM® SPSS® Statistics procedure to analyze your IBM SPSS Modeler data. A wide variety of IBM SPSS Statistics analytical procedures is available. This node requires a licensed copy of IBM SPSS Statistics.
---	--

Example

```

stream = modeler.script.stream()
statisticsoutputnode = stream.createAt("statisticsoutput", "Output", 200, 200)
statisticsoutputnode.setPropertyValue("syntax", "SORT CASES BY Age(A) Sex(A) BP(A) Cholesterol(A) ")
statisticsoutputnode.setPropertyValue("use_output_name", False)
statisticsoutputnode.setPropertyValue("output_mode", "File")
statisticsoutputnode.setPropertyValue("full_filename", "Cases by Age, Sex and Medical History")
statisticsoutputnode.setPropertyValue("file_type", "HTML")

```

Table 1. statisticsoutputnode properties

statisticsoutputnode properties	Data type	Property description
mode	Dialog Syntax	Selects "IBM SPSS Statistics dialog" option or Syntax Editor
syntax	<i>string</i>	
use_output_name	<i>flag</i>	
output_name	<i>string</i>	
output_mode	Screen File	
full_filename	<i>string</i>	
file_type	HTML SPV SPW	

Related information

- [Node properties overview](#)

- [featureselectionnode properties](#)
- [Output node properties](#)
- [analysisnode properties](#)
- [dataauditnode properties](#)
- [matrixnode properties](#)
- [meansnode properties](#)
- [reportnode properties](#)
- [setglobalsnode properties](#)
- [statisticsnode properties](#)
- [tablenode properties](#)
- [transformnode properties](#)
- [databaseexportnode properties](#)
- [datacollectionexportnode Properties](#)
- [excelexportnode Properties](#)
- [outputfilenode Properties](#)
- [publishernode Properties](#)
- [sasexportnode Properties](#)
- [xmlexportnode Properties](#)
- [statisticsexportnode Properties](#)

statisticsexportnode Properties



The Statistics Export node outputs data in IBM® SPSS® Statistics .sav or .zsav format. The .sav or .zsav files can be read by IBM SPSS Statistics Base and other products. This is also the format used for cache files in IBM SPSS Modeler.

Example

```
stream = modeler.script.stream()
statisticsexportnode = stream.createAt("statisticsexport", "Export", 200, 200)
statisticsexportnode.setPropertyValue("full_filename", "c:/output/SPSS_Statistics_out.sav")
statisticsexportnode.setPropertyValue("field_names", "Names")
statisticsexportnode.setPropertyValue("launch_application", True)
statisticsexportnode.setPropertyValue("generate_import", True)
```

Table 1. statisticsexportnode properties

statisticsexportnode properties	Data type	Property description
full_filename	string	
file_type	sav zsav	Save file in sav or zsav format. For example: <code>statisticsexportnode.setPropertyValue("file_type", "sav")</code>
encrypt_file	flag	Whether or not the file is password protected.
password	string	The password.
launch_application	flag	
export_names	NamesAndLabels NamesAsLabels	Used to map field names from IBM SPSS Modeler upon export to IBM SPSS Statistics or SAS variable names.
generate_import	flag	

Related information

- [Node properties overview](#)
- [featureselectionnode properties](#)
- [Output node properties](#)
- [analysisnode properties](#)
- [dataauditnode properties](#)
- [matrixnode properties](#)
- [meansnode properties](#)
- [reportnode properties](#)
- [setglobalsnode properties](#)
- [statisticsnode properties](#)
- [tablenode properties](#)
- [transformnode properties](#)
- [databaseexportnode properties](#)
- [datacollectionexportnode Properties](#)
- [excelexportnode Properties](#)
- [outputfilenode Properties](#)

- [publishernode Properties](#)
- [sasexportnode Properties](#)
- [xmlexportnode Properties](#)
- [statisticsoutputnode Properties](#)

Python Node Properties

- [gmm properties](#)
- [hdbscannode properties](#)
- [kdemodel properties](#)
- [kdeexport properties](#)
- [ocsvmnode properties](#)
- [rfnode properties](#)
- [smotenode Properties](#)
- [tsnenode Properties](#)
- [xgboostlinearnode Properties](#)
- [xgboosttreenode Properties](#)

gmm properties



A Gaussian Mixture© model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians. The Gaussian Mixture node in SPSS® Modeler exposes the core features and commonly used parameters of the Gaussian Mixture library. The node is implemented in Python.

Table 1. gmm properties

gmm properties	Data type	Property description
<code>use_partition</code>	<code>boolean</code>	Set to <code>True</code> or <code>False</code> to specify whether to use partitioned data. Default is <code>False</code> .
<code>covariance_type</code>	<code>string</code>	Specify <code>Full</code> , <code>Tied</code> , <code>Diag</code> , or <code>Spherical</code> to set the covariance type.
<code>number_component</code>	<code>integer</code>	Specify an integer for the number of mixture components. Minimum value is 1. Default value is 2.
<code>component_label</code>	<code>boolean</code>	Specify <code>True</code> to set the cluster label to a string or <code>False</code> to set the cluster label to a number. Default is <code>False</code> .
<code>label_prefix</code>	<code>string</code>	If using a string cluster label, you can specify a prefix.
<code>enable_random_seed</code>	<code>boolean</code>	Specify <code>True</code> if you want to use a random seed. Default is <code>False</code> .
<code>random_seed</code>	<code>integer</code>	If using a random seed, specify an integer to be used for generating random samples.
<code>tol</code>	<code>Double</code>	Specify the convergence threshold. Default is <code>0.000_1</code> .
<code>max_iter</code>	<code>integer</code>	Specify the maximum number of iterations to perform. Default is 100.
<code>init_params</code>	<code>string</code>	Set the initialization parameter to use. Options are <code>Kmeans</code> or <code>Random</code> .
<code>warm_start</code>	<code>boolean</code>	Specify <code>True</code> to use the solution of the last fitting as the initialization for the next call of fit. Default is <code>False</code> .

hdbscannode properties



Hierarchical Density-Based Spatial Clustering (HDBSCAN)© uses unsupervised learning to find clusters, or dense regions, of a data set. The HDBSCAN node in SPSS® Modeler exposes the core features and commonly used parameters of the HDBSCAN library. The node is implemented in Python, and you can use it to cluster your dataset into distinct groups when you don't know what those groups are at first.

Table 1. hdbscannode properties

hdbscannode properties	Data type	Property description
<code>inputs</code>	<code>field</code>	Input fields for clustering.
<code>useHPO</code>	<code>boolean</code>	Specify <code>true</code> or <code>false</code> to enable or disable Hyper-Parameter Optimization (HPO) based on Rbfopt, which automatically discovers the optimal combination of parameters so that the model will achieve the expected or lower error rate on the samples. Default is <code>false</code> .
<code>min_cluster_size</code>	<code>integer</code>	The minimum size of clusters. Specify an integer. Default is 5.

hdbscannode properties	Data type	Property description
<code>min_samples</code>	<code>integer</code>	The number of samples in a neighborhood for a point to be considered a core point. Specify an integer. If set to 0, the <code>min_cluster_size</code> is used. Default is 0.
<code>algorithm</code>	<code>string</code>	Specify which algorithm to use: <code>best</code> , <code>generic</code> , <code>prims_kdtree</code> , <code>prims_balltree</code> , <code>boruvka_kdtree</code> , or <code>boruvka_balltree</code> . Default is <code>best</code> .
<code>metric</code>	<code>string</code>	Specify which metric to use when calculating distance between instances in a feature array: <code>euclidean</code> , <code>cityblock</code> , <code>L1</code> , <code>L2</code> , <code>manhattan</code> , <code>braycurtis</code> , <code>canberra</code> , <code>chebyshev</code> , <code>correlation</code> , <code>minkowski</code> , or <code>squareeuclidean</code> . Default is <code>euclidean</code> .
<code>useStringLabel</code>	<code>boolean</code>	Specify <code>true</code> to use a string cluster label, or <code>false</code> to use a number cluster label. Default is <code>false</code> .
<code>stringLabelPrefix</code>	<code>string</code>	If the <code>useStringLabel</code> parameter is set to <code>true</code> , specify a value for the string label prefix. Default prefix is <code>cluster</code> .
<code>approx_min_span_tree</code>	<code>boolean</code>	Specify <code>true</code> to accept an approximate minimum spanning tree, or <code>false</code> if you are willing to sacrifice speed for correctness. Default is <code>true</code> .
<code>cluster_selection_method</code>	<code>string</code>	Specify the method to use for selecting clusters from the condensed tree: <code>eom</code> or <code>leaf</code> . Default is <code>eom</code> (Excess of Mass algorithm).
<code>allow_single_cluster</code>	<code>boolean</code>	Specify <code>true</code> if you want to allow single cluster results. Default is <code>false</code> .
<code>p_value</code>	<code>double</code>	Specify the <code>p_value</code> to use if you're using <code>minkowski</code> for the metric. Default is 1.5.
<code>leaf_size</code>	<code>integer</code>	If using a space tree algorithm (<code>boruvka_kdtree</code> , or <code>boruvka_balltree</code>), specify the number of points in a leaf node of the tree. Default is 40.
<code>outputValidity</code>	<code>boolean</code>	Specify <code>true</code> or <code>false</code> to control whether the Validity Index chart is included in the model output.
<code>outputCondensed</code>	<code>boolean</code>	Specify <code>true</code> or <code>false</code> to control whether the Condensed Tree chart is included in the model output.
<code>outputSingleLinkage</code>	<code>boolean</code>	Specify <code>true</code> or <code>false</code> to control whether the Single Linkage Tree chart is included in the model output.
<code>outputMinSpan</code>	<code>boolean</code>	Specify <code>true</code> or <code>false</code> to control whether the Min Span Tree chart is included in the model output.
<code>is_split</code>		Added in version 18.2.1.1.

kdemodel properties



Kernel Density Estimation (KDE)© uses the Ball Tree or KD Tree algorithms for efficient queries, and combines concepts from unsupervised learning, feature engineering, and data modeling. Neighbor-based approaches such as KDE are some of the most popular and useful density estimation techniques. The KDE Modeling and KDE Simulation nodes in SPSS® Modeler expose the core features and commonly used parameters of the KDE library. The nodes are implemented in Python.

Table 1. kdemodel properties

kdemodel properties	Data type	Property description
<code>bandwidth</code>	<code>double</code>	Default is 1.
<code>kernel</code>	<code>string</code>	The kernel to use: <code>gaussian</code> , <code>tophat</code> , <code>epanechnikov</code> , <code>exponential</code> , <code>linear</code> , or <code>cosine</code> . Default is <code>gaussian</code> .
<code>algorithm</code>	<code>string</code>	The tree algorithm to use: <code>kd_tree</code> , <code>ball_tree</code> , or <code>auto</code> . Default is <code>auto</code> .
<code>metric</code>	<code>string</code>	The metric to use when calculating distance. For the <code>kd_tree</code> algorithm, choose from: <code>Euclidean</code> , <code>Chebyshev</code> , <code>Cityblock</code> , <code>Minkowski</code> , <code>Manhattan</code> , <code>Infinity</code> , <code>P</code> , <code>L2</code> , or <code>L1</code> . For the <code>ball_tree</code> algorithm, choose from: <code>Euclidian</code> , <code>Braycurtis</code> , <code>Chebyshev</code> , <code>Canberra</code> , <code>Cityblock</code> , <code>Dice</code> , <code>Hamming</code> , <code>Infinity</code> , <code>Jaccard</code> , <code>L1</code> , <code>L2</code> , <code>Minkowski</code> , <code>Matching</code> , <code>Manhattan</code> , <code>P</code> , <code>Rogersanimoto</code> , <code>Russellrao</code> , <code>Sokalmichener</code> , <code>Sokalsneath</code> , or <code>Kulsinski</code> . Default is <code>Euclidean</code> .
<code>atol</code>	<code>float</code>	The desired absolute tolerance of the result. A larger tolerance will generally lead to faster execution. Default is 0.0.
<code>rtol</code>	<code>float</code>	The desired relative tolerance of the result. A larger tolerance will generally lead to faster execution. Default is <code>1E-8</code> .
<code>breadthFirst</code> renamed to <code>breadth_first</code> starting with version 18.2.1.1	<code>boolean</code>	Set to <code>True</code> to use a breadth-first approach. Set to <code>False</code> to use a depth-first approach. Default is <code>True</code> .
<code>LeafSize</code> renamed to <code>leaf_size</code> starting with version 18.2.1.1	<code>integer</code>	The leaf size of the underlying tree. Default is 40. Changing this value may significantly impact the performance.

kdemodel properties	Data type	Property description
pValue	double	Specify the P Value to use if you're using Minkowski for the metric. Default is 1.5 .
custom_name		
default_node_name		
use_HPO		

kdeexport properties

	Kernel Density Estimation (KDE)© uses the Ball Tree or KD Tree algorithms for efficient queries, and combines concepts from unsupervised learning, feature engineering, and data modeling. Neighbor-based approaches such as KDE are some of the most popular and useful density estimation techniques. The KDE Modeling and KDE Simulation nodes in SPSS® Modeler expose the core features and commonly used parameters of the KDE library. The nodes are implemented in Python.
---	---

Table 1. kdeexport properties

kdeexport properties	Data type	Property description
bandwidth	double	Default is 1 .
kernel	string	The kernel to use: gaussian or tophat . Default is gaussian .
algorithm	string	The tree algorithm to use: kd_tree , ball_tree , or auto . Default is auto .
metric	string	The metric to use when calculating distance. For the kd_tree algorithm, choose from: Euclidean , Chebyshev , Cityblock , Minkowski , Manhattan , Infinity , P , L2 , or L1 . For the ball_tree algorithm, choose from: Euclidian , Braycurtis , Chebyshev , Canberra , Cityblock , Dice , Hamming , Infinity , Jaccard , L1 , L2 , Minkowski , Matching , Manhattan , P , Rogersanimoto , Russellrao , Sokalmichener , Sokalsneath , or Kulsinski . Default is Euclidean .
atol	float	The desired absolute tolerance of the result. A larger tolerance will generally lead to faster execution. Default is 0.0 .
rtol	float	The desired relative tolerance of the result. A larger tolerance will generally lead to faster execution. Default is 1E-8 .
breadthFirst	boolean	Set to True to use a breadth-first approach. Set to False to use a depth-first approach. Default is True .
LeafSize	integer	The leaf size of the underlying tree. Default is 40 . Changing this value may significantly impact the performance.
pValue	double	Specify the P Value to use if you're using Minkowski for the metric. Default is 1.5 .

gmm properties

	A Gaussian Mixture© model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians. The Gaussian Mixture node in SPSS® Modeler exposes the core features and commonly used parameters of the Gaussian Mixture library. The node is implemented in Python.
---	---

Table 1. gmm properties

gmm properties	Data type	Property description
use_partition	boolean	Set to True or False to specify whether to use partitioned data. Default is False .
covariance_type	string	Specify Full , Tied , Diag , or Spherical to set the covariance type.
number_component	integer	Specify an integer for the number of mixture components. Minimum value is 1 . Default value is 2 .
component_label	boolean	Specify True to set the cluster label to a string or False to set the cluster label to a number. Default is False .
label_prefix	string	If using a string cluster label, you can specify a prefix.
enable_random_seed	boolean	Specify True if you want to use a random seed. Default is False .
random_seed	integer	If using a random seed, specify an integer to be used for generating random samples.
tol	Double	Specify the convergence threshold. Default is 0.0001 .
max_iter	integer	Specify the maximum number of iterations to perform. Default is 100 .
init_params	string	Set the initialization parameter to use. Options are Kmeans or Random .
warm_start	boolean	Specify True to use the solution of the last fitting as the initialization for the next call of fit. Default is False .

ocsvmnode properties



The One-Class SVM node uses an unsupervised learning algorithm. The node can be used for novelty detection. It will detect the soft boundary of a given set of samples, to then classify new points as belonging to that set or not. This One-Class SVM modeling node in SPSS® Modeler is implemented in Python and requires the scikit-learn© Python library.

Table 1. ocsvmnode properties

ocsvmnode properties	Data type	Property description
role_use Renamed to custom_fields starting with version 18.2.1.1	string	Specify predefined to use predefined roles or custom to use custom field assignments. Default is predefined.
splits	field	List of the field names for split.
use_partition	Boolean	Specify true or false . Default is true . If set to true , only training data will be used when building the model.
mode_type	string	The mode. Possible values are simple or expert . All parameters on the Expert tab will be disabled if simple is specified.
stopping_criteria	string	A string of scientific notation. Possible values are 1.0E-1 , 1.0E-2 , 1.0E-3 , 1.0E-4 , 1.0E-5 , or 1.0E-6 . Default is 1.0E-3 .
precision	float	The regression precision (<i>nu</i>). Bound on the fraction of training errors and support vectors. Specify a number greater than 0 and less than or equal to 1.0. Default is 0.1.
kernel	string	The kernel type to use in the algorithm. Possible values are linear , poly , rbf , sigmoid , or precomputed . Default is rbf .
enable_gamma	Boolean	Enables the gamma parameter. Specify true or false . Default is true .
gamma	float	This parameter is only enabled for the kernels rbf , poly , and sigmoid . If the enable_gamma parameter is set to false , this parameter will be set to auto . If set to true , the default is 0.1.
coef0	float	Independent term in the kernel function. This parameter is only enabled for the poly kernel and the sigmoid kernel. Default value is 0.0.
degree	integer	Degree of the polynomial kernel function. This parameter is only enabled for the poly kernel. Specify any integer. Default is 3.
shrinking	Boolean	Specifies whether to use the shrinking heuristic option. Specify true or false . Default is false .
enable_cache_size	Boolean	Enables the cache_size parameter. Specify true or false . Default is false .
cache_size	float	The size of the kernel cache in MB. Default is 200.
pc_type	string	The type of the parallel coordinates graphic. Possible options are independent or general .
lines_amount	integer	Maximum number of lines to include on the graphic. Specify an integer between 1 and 1000.
lines_fields_custom	Boolean	Enables the lines_fields parameter, which allows you to specify custom fields to show in the graph output. If set to false , all fields will be shown. If set to true , only the fields specified with the lines_fields parameter will be shown. For performance reasons, a maximum of 20 fields will be displayed.
lines_fields	field	List of the field names to include on the graphic as vertical axes.
enable_graphic	Boolean	Specify true or false . Enables graphic output (disable this option if you want to save time and reduce stream file size).
enable_hpo	Boolean	Specify true or false to enable or disable the HPO options. If set to true , Rbfopt will be applied to find out the "best" One-Class SVM model automatically, which reaches the target objective value defined by the user with the following target_objval parameter.
target_objval	float	The objective function value (error rate of the model on the samples) we want to reach (for example, the value of the unknown optimum). Set this parameter to the appropriate value if the optimum is unknown (for example, 0.01).
max_iterations	integer	Maximum number of iterations for trying the model. Default is 1000.
max_evaluations	integer	Maximum number of function evaluations for trying the model, where the focus is accuracy over speed. Default is 300.

rfnode properties



The Random Forest node uses an advanced implementation of a bagging algorithm with a tree model as the base model. This Random Forest modeling node in SPSS® Modeler is implemented in Python and requires the scikit-learn© Python library.

Table 1. rfnode properties

rfnode properties	Data type	Property description
<code>role_use</code>	<code>string</code>	Specify <code>predefined</code> to use predefined roles or <code>custom</code> to use custom field assignments. Default is <code>predefined</code> .
<code>inputs</code>	<code>field</code>	List of the field names for input.
<code>splits</code>	<code>field</code>	List of the field names for split.
<code>n_estimators</code>	<code>integer</code>	Number of trees to build. Default is <code>10</code> .
<code>specify_max_depth</code>	<code>Boolean</code>	Specify custom max depth. If <code>false</code> , nodes are expanded until all leaves are pure or until all leaves contain less than <code>min_samples_split</code> samples. Default is <code>false</code> .
<code>max_depth</code>	<code>integer</code>	The maximum depth of the tree. Default is <code>10</code> .
<code>min_samples_leaf</code>	<code>integer</code>	Minimum leaf node size. Default is <code>1</code> .
<code>max_features</code>	<code>string</code>	<p>The number of features to consider when looking for the best split:</p> <ul style="list-style-type: none"> If <code>auto</code>, then <code>max_features=sqrt(n_features)</code> for classifier and <code>max_features=sqrt(n_features)</code> for regression. If <code>sqrt</code>, then <code>max_features=sqrt(n_features)</code>. If <code>log2</code>, then <code>max_features=log2 (n_features)</code>. <p>Default is <code>auto</code>.</p>
<code>bootstrap</code>	<code>Boolean</code>	Use bootstrap samples when building trees. Default is <code>true</code> .
<code>oob_score</code>	<code>Boolean</code>	Use out-of-bag samples to estimate the generalization accuracy. Default value is <code>false</code> .
<code>extreme</code>	<code>Boolean</code>	Use extremely randomized trees. Default is <code>false</code> .
<code>use_random_seed</code>	<code>Boolean</code>	Specify this to get replicated results. Default is <code>false</code> .
<code>random_seed</code>	<code>integer</code>	The random number seed to use when build trees. Specify any integer.
<code>cache_size</code>	<code>float</code>	The size of the kernel cache in MB. Default is <code>200</code> .
<code>enable_random_seed</code>	<code>Boolean</code>	Enables the <code>random_seed</code> parameter. Specify true or false. Default is <code>false</code> .
<code>enable_hpo</code>	<code>Boolean</code>	Specify <code>true</code> or <code>false</code> to enable or disable the HPO options. If set to <code>true</code> , Rbfopt will be applied to determine the "best" Random Forest model automatically, which reaches the target objective value defined by the user with the following <code>target_objval</code> parameter.
<code>target_objval</code>	<code>float</code>	The objective function value (error rate of the model on the samples) you want to reach (for example, the value of the unknown optimum). Set this parameter to the appropriate value if the optimum is unknown (for example, <code>0.01</code>).
<code>max_iterations</code>	<code>integer</code>	Maximum number of iterations for trying the model. Default is <code>1000</code> .
<code>max_evaluations</code>	<code>integer</code>	Maximum number of function evaluations for trying the model, where the focus is accuracy over speed. Default is <code>300</code> .

smotenode Properties



The Synthetic Minority Over-sampling Technique (SMOTE) node provides an over-sampling algorithm to deal with imbalanced data sets. It provides an advanced method for balancing data. The SMOTE process node in SPSS® Modeler is implemented in Python and requires the imbalanced-learn© Python library.

Table 1. smotenode properties

smotenode properties	Data type	Property description
<code>target_field</code> Renamed to <code>target</code> starting with version 18.2.1.1	<code>field</code>	The target field.
<code>sample_ratio</code>	<code>string</code>	Enables a custom ratio value. The two options are Auto (<code>sample_ratio_auto</code>) or Set ratio (<code>sample_ratio_manual</code>).
<code>sample_ratio_value</code>	<code>float</code>	The ratio is the number of samples in the minority class over the number of samples in the majority class. It must be larger than <code>0</code> and less than or equal to <code>1</code> . Default is <code>auto</code> .
<code>enable_random_seed</code>	<code>Boolean</code>	If set to <code>true</code> , the <code>random_seed</code> property will be enabled.
<code>random_seed</code>	<code>integer</code>	The seed used by the random number generator.
<code>k_neighbours</code>	<code>integer</code>	The number of nearest neighbours to be used for constructing synthetic samples. Default is <code>5</code> .
<code>m_neighbours</code>	<code>integer</code>	The number of nearest neighbours to be used for determining if a minority sample is in danger. This option is only enabled with the SMOTE algorithm types <code>borderline1</code> and <code>borderline2</code> . Default is <code>10</code> .

smotenode properties	Data type	Property description
algorithm_kind Renamed to algorithm starting with version 18.2.1.1	<i>string</i>	The type of SMOTE algorithm: <code>regular</code> , <code>borderline1</code> , or <code>borderline2</code> .
usepartition Renamed to use_partition starting with version 18.2.1.1	<i>Boolean</i>	If set to <code>true</code> , only training data will be used for model building. Default is <code>true</code> .

tsnenode Properties



t-Distributed Stochastic Neighbor Embedding (t-SNE) is a tool for visualizing high-dimensional data. It converts affinities of data points to probabilities. This t-SNE node in SPSS® Modeler is implemented in Python and requires the `scikit-learn`® Python library.

Table 1. tsnenode properties

tsnenode properties	Data type	Property description
mode_type	<i>string</i>	Specify <code>simple</code> or <code>expert</code> mode.
n_components	<i>string</i>	Dimension of the embedded space (2D or 3D). Specify <code>2</code> or <code>3</code> . Default is <code>2</code> .
method	<i>string</i>	Specify <code>barnes_hut</code> or <code>exact</code> . Default is <code>barnes_hut</code> .
init	<i>string</i>	Initialization of embedding. Specify <code>random</code> or <code>pca</code> . Default is <code>random</code> .
target_field Renamed to target starting with version 18.2.1.1	<i>string</i>	Target field name. It will be a colormap on the output graph. The graph will use one color if no target field is specified.
perplexity	<i>float</i>	The perplexity is related to the number of nearest neighbors used in other manifold learning algorithms. Larger datasets usually require a larger perplexity. Consider selecting a value between 5 and 50. Default is 30.
early_exaggeration	<i>float</i>	Controls how tight the natural clusters in the original space are in the embedded space, and how much space will be between them. Default is 12.0.
learning_rate	<i>float</i>	Default is 200.
n_iter	<i>integer</i>	Maximum number of iterations for the optimization. Set to at least 250. Default is 1000.
angle	<i>float</i>	The angular size of the distant node as measured from a point. Specify a value in the range of 0-1. Default is 0.5.
enable_random_seed	<i>Boolean</i>	Set to <code>true</code> to enable the <code>random_seed</code> parameter. Default is <code>false</code> .
random_seed	<i>integer</i>	The random number seed to use. Default is <code>None</code> .
n_iter_without_progress	<i>integer</i>	Maximum iterations without progress. Default is 300.
min_grad_norm	<i>string</i>	If the gradient norm is below this threshold, the optimization will be stopped. Default is <code>1.0E-7</code> . Possible values are: <ul style="list-style-type: none"> • <code>1.0E-1</code> • <code>1.0E-2</code> • <code>1.0E-3</code> • <code>1.0E-4</code> • <code>1.0E-5</code> • <code>1.0E-6</code> • <code>1.0E-7</code> • <code>1.0E-8</code>
isGridSearch	<i>Boolean</i>	Set to <code>true</code> to perform t-SNE with several different perplexities. Default is <code>false</code> .
output_Rename	<i>Boolean</i>	Specify <code>true</code> if you want to provide a custom name, or <code>false</code> to name the output automatically. Default is <code>false</code> .
output_to	<i>string</i>	Specify <code>Screen</code> or <code>Output</code> . Default is <code>Screen</code> .
full_filename	<i>string</i>	Specify the output file name.
output_file_type	<i>string</i>	Output file format. Specify <code>HTML</code> or <code>Output object</code> . Default is <code>HTML</code> .

xgboostlinearnode Properties



XGBoost Linear® is an advanced implementation of a gradient boosting algorithm with a linear model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. The XGBoost Linear node in SPSS® Modeler is implemented in Python.

Table 1. xgboostlinearnode properties

xgboostlinearnode properties	Data type	Property description
TargetField Renamed to target starting with version 18.2.1.1	<i>field</i>	
InputFields Renamed to inputs starting with version 18.2.1.1	<i>field</i>	
alpha	<i>Double</i>	The alpha linear booster parameter. Specify any number 0 or greater. Default is 0.
lambda	<i>Double</i>	The lambda linear booster parameter. Specify any number 0 or greater. Default is 1.
lambdaBias	<i>Double</i>	The lambda bias linear booster parameter. Specify any number. Default is 0.
numBoostRound Renamed to num_boost_round starting with version 18.2.1.1	<i>integer</i>	The num boost round value for model building. Specify a value between 1 and 1000. Default is 10.
objectiveType	<i>string</i>	The objective type for the learning task. Possible values are reg:linear , reg:logistic , reg:gamma , reg:tweedie , count:poisson , rank:pairwise , binary:logistic , or multi . Note that for flag targets, only binary:logistic or multi can be used. If multi is used, the score result will show the multi:softmax and multi:softprob XGBoost objective types.
random_seed	<i>integer</i>	The random number seed. Any number between 0 and 9999999. Default is 0.
useHPO	<i>Boolean</i>	Specify true or false to enable or disable the HPO options. If set to true , Rbfopt will be applied to find out the "best" One-Class SVM model automatically, which reaches the target objective value defined by the user with the target_objval parameter.

xgboosttreenode Properties



XGBoost Tree® is an advanced implementation of a gradient boosting algorithm with a tree model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost Tree is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost Tree node in SPSS® Modeler exposes the core features and commonly used parameters. The node is implemented in Python.

Table 1. xgboosttreenode properties

xgboosttreenode properties	Data type	Property description
TargetField Renamed to target starting with version 18.2.1.1	<i>field</i>	The target fields.
InputFields Renamed to inputs starting with version 18.2.1.1	<i>field</i>	The input fields.
treeMethod Renamed to tree_method starting with version 18.2.1.1	<i>string</i>	The tree method for model building. Possible values are auto , exact , or approx . Default is auto .
numBoostRound Renamed to num_boost_round starting with version 18.2.1.1	<i>integer</i>	The num boost round value for model building. Specify a value between 1 and 1000. Default is 10.
maxDepth Renamed to max_depth starting with version 18.2.1.1	<i>integer</i>	The max depth for tree growth. Specify a value of 1 or higher. Default is 6.
minChildWeight Renamed to min_child_weight starting with version 18.2.1.1	<i>Double</i>	The min child weight for tree growth. Specify a value of 0 or higher. Default is 1.
maxDeltaStep Renamed to max_delta_step starting with version 18.2.1.1	<i>Double</i>	The max delta step for tree growth. Specify a value of 0 or higher. Default is 0.

xgboosttreenode properties	Data type	Property description
objectiveType Renamed to objective_type starting with version 18.2.1.1	<i>string</i>	The objective type for the learning task. Possible values are reg: linear , reg: logistic , reg: gamma , reg: tweedie , count: poisson , rank: pairwise , binary: logistic , or multi . Note that for flag targets, only binary: logistic or multi can be used. If multi is used, the score result will show the multi: softmax and multi: softprob XGBoost objective types.
earlyStopping Renamed to early_stopping starting with version 18.2.1.1	<i>Boolean</i>	Whether to use the early stopping function. Default is False .
earlyStoppingRounds Renamed to early_stopping_rounds starting with version 18.2.1.1	<i>integer</i>	Validation error needs to decrease at least every early stopping round(s) to continue training. Default is 10 .
evaluationDataRatio Renamed to evaluation_data_ratio starting with version 18.2.1.1	<i>Double</i>	Ration of input data used for validation errors. Default is 0 . 3 .
random_seed	<i>integer</i>	The random number seed. Any number between 0 and 9999999 . Default is 0 .
sampleSize Renamed to sample_size starting with version 18.2.1.1	<i>Double</i>	The sub sample for control overfitting. Specify a value between 0 . 1 and 1 . 0 . Default is 0 . 1 .
eta	<i>Double</i>	The eta for control overfitting. Specify a value between 0 and 1 . Default is 0 . 3 .
gamma	<i>Double</i>	The gamma for control overfitting. Specify any number 0 or greater. Default is 6 .
colsSampleRatio Renamed to col_sample_ratio starting with version 18.2.1.1	<i>Double</i>	The colsample by tree for control overfitting. Specify a value between 0 . 01 and 1 . Default is 1 .
colsSampleLevel Renamed to col_sample_level starting with version 18.2.1.1	<i>Double</i>	The colsample by level for control overfitting. Specify a value between 0 . 01 and 1 . Default is 1 .
lambda	<i>Double</i>	The lambda for control overfitting. Specify any number 0 or greater. Default is 1 .
alpha	<i>Double</i>	The alpha for control overfitting. Specify any number 0 or greater. Default is 0 .
scalePosWeight Renamed to scale_pos_weight starting with version 18.2.1.1	<i>Double</i>	The scale pos weight for handling imbalanced datasets. Default is 1 .
use_HPO Added for version 18.2.1.1		

Spark Node Properties

- [isotonicasnode Properties](#)
- [kmeansasnodel properties](#)
- [multilayerperceptronnode Properties](#)
- [xgboostasnodel Properties](#)

isotonicasnode Properties



Isotonic Regression belongs to the family of regression algorithms. The Isotonic-AS node in SPSS® Modeler is implemented in Spark. For details about Isotonic Regression algorithms, see <https://spark.apache.org/docs/2.2.0/mllib-isotonic-regression.html>.

Table 1. isotonicasnode properties

isotonicasnode properties	Data type	Property description
----------------------------------	------------------	-----------------------------

isotonicasnode properties	Data type	Property description
label	string	This property is a dependent variable for which isotonic regression is calculated.
features	string	This property is an independent variable.
weightCol	string	The weight represents a number of measures. Default is 1.
isotonic	Boolean	This property indicates whether the type is isotonic or antitonic .
featureIndex	integer	This property is for the index of the feature if featuresCol is a vector column. Default is 0.

kmeansasnode properties



K-Means is one of the most commonly used clustering algorithms. It clusters data points into a predefined number of clusters. The K-Means-AS node in SPSS® Modeler is implemented in Spark. For details about K-Means algorithms, see <https://spark.apache.org/docs/2.2.0/ml-clustering.html>. Note that the K-Means-AS node performs one-hot encoding automatically for categorical variables.

Table 1. kmeansasnode properties

kmeansasnode Properties	Values	Property description
roleUse	string	Specify predefined to use predefined roles, or custom to use custom field assignments. Default is predefined .
autoModel	Boolean	Specify true to use the default name (\$S-prediction) for the new generated scoring field, or false to use a custom name. Default is true .
features	field	List of the field names for input when the roleUse property is set to custom .
name	string	The name of the new generated scoring field when the autoModel property is set to false .
clustersNum	integer	The number of clusters to create. Default is 5.
initMode	string	The initialization algorithm. Possible values are k-means or random . Default is k-means .
initSteps	integer	The number of initialization steps when initMode is set to k-means . Default is 2.
advancedSettings	Boolean	Specify true to make the following four properties available. Default is false .
maxIteration	integer	Maximum number of iterations for clustering. Default is 20.
tolerance	string	The tolerance to stop the iterations. Possible settings are 1.0E-1, 1.0E-2, ..., 1.0E-6 . Default is 1.0E-4 .
setSeed	Boolean	Specify true to use a custom random seed. Default is false .
randomSeed	integer	The custom random seed when the setSeed property is true .

multilayerperceptronnode Properties



Multilayer perceptron is a classifier based on the feedforward artificial neural network and consists of multiple layers. Each layer is fully connected to the next layer in the network. The MultiLayerPerceptron-AS node in SPSS® Modeler is implemented in Spark. For details about the multilayer perceptron classifier (MLPC), see <https://spark.apache.org/docs/latest/ml-classification-regression.html#multilayer-perceptron-classifier>.

Table 1. multilayerperceptronnode properties

multilayerperceptronnode properties	Data type	Property description
features	field	One or more fields to use as inputs for the prediction.
label	field	The field to use as the target for the prediction.
layers[0]	integer	The number of perceptron layers to include. Default is 1.
layers[1...<latest-1>]	integer	The number of hidden layers. Default is 1.
layers[<latest>]	integer	The number of output layers. Default is 1.
seed	integer	The custom random seed.
maxiter	integer	The maximum number of iterations to perform. Default is 10.

xgboostasnode Properties



XGBoost is an advanced implementation of a gradient boosting algorithm. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost-AS node in SPSS® Modeler exposes the core features and commonly used parameters. The XGBoost-AS node is implemented in Spark.

Table 1. xgboostasnode properties

xgboostasnode properties	Data type	Property description
target_field	<i>field</i>	List of the field names for target.
input_fields	<i>field</i>	List of the field names for inputs.
nWorkers	<i>integer</i>	The number of workers used to train the XGBoost model. Default is 1.
numThreadPerTask	<i>integer</i>	The number of threads used per worker. Default is 1.
useExternalMemory	<i>Boolean</i>	Whether to use external memory as cache. Default is false.
boosterType	<i>string</i>	The booster type to use. Available options are gbtree , gblinear , or dart . Default is gbtree .
numBoostRound	<i>integer</i>	The number of rounds for boosting. Specify a value of 0 or higher. Default is 10.
scalePosWeight	<i>Double</i>	Control the balance of positive and negative weights. Default is 1.
randomseed	<i>integer</i>	The seed used by the random number generator. Default is 0.
objectiveType	<i>string</i>	The learning objective. Possible values are reg:linear , reg:logistic , reg:gamma , reg:tweedie , rank:pairwise , binary:logistic , or multi . Note that for flag targets, only binary:logistic or multi can be used. If multi is used, the score result will show the multi:softmax and multi:softprob XGBoost objective types. Default is reg:linear .
evalMetric	<i>string</i>	Evaluation metrics for validation data. A default metric will be assigned according to the objective. Possible values are rmse , mae , logloss , error , merror , mlogloss , auc , ndcg , map , or gamma-deviance . Default is rmse .
lambda	<i>Double</i>	L2 regularization term on weights. Increasing this value will make the model more conservative. Specify any number 0 or greater. Default is 1.
alpha	<i>Double</i>	L1 regularization term on weights. Increasing this value will make the model more conservative. Specify any number 0 or greater. Default is 0.
lambdaBias	<i>Double</i>	L2 regularization term on bias. If the gblinear booster type is used, this lambda bias linear booster parameter is available. Specify any number 0 or greater. Default is 0.
treeMethod	<i>string</i>	If the gbtree or dart booster type is used, this tree method parameter for tree growth (and the other tree parameters that follow) is available. It specifies the XGBoost tree construction algorithm to use. Available options are auto , exact , or approx . Default is auto .
maxDepth	<i>integer</i>	The maximum depth for trees. Specify a value of 2 or higher. Default is 6.
minChildWeight	<i>Double</i>	The minimum sum of instance weight (hessian) needed in a child. Specify a value of 0 or higher. Default is 1.
maxDeltaStep	<i>Double</i>	The maximum delta step to allow for each tree's weight estimation. Specify a value of 0 or higher. Default is 0.
sampleSize	<i>Double</i>	The sub sample for is the ratio of the training instance. Specify a value between 0.1 and 1.0. Default is 1.0.
eta	<i>Double</i>	The step size shrinkage used during the update step to prevent overfitting. Specify a value between 0 and 1. Default is 0.3.
gamma	<i>Double</i>	The minimum loss reduction required to make a further partition on a leaf node of the tree. Specify any number 0 or greater. Default is 6.
colSampleRatio	<i>Double</i>	The sub sample ratio of columns when constructing each tree. Specify a value between 0.01 and 1. Default is 1.
colSampleLevel	<i>Double</i>	The sub sample ratio of columns for each split, in each level. Specify a value between 0.01 and 1. Default is 1.
normalizeType	<i>string</i>	If the dart booster type is used, this dart parameter and the following three dart parameters are available. This parameter sets the normalization algorithm. Specify tree or forest . Default is tree .
sampleType	<i>string</i>	The sampling algorithm type. Specify uniform or weighted . Default is uniform .
rateDrop	<i>Double</i>	The dropout rate dart booster parameter. Specify a value between 0.0 and 1.0. Default is 0.0.
skipDrop	<i>Double</i>	The dart booster parameter for the probability of skip dropout. Specify a value between 0.0 and 1.0. Default is 0.0.

SuperNode properties

Properties that are specific to SuperNodes are described in the following tables. Note that common node properties also apply to SuperNodes.

Table 1. Terminal supernode properties

Property name	Property type/List of values	Property description
execute_method	Script Normal	

Property name	Property type/List of values	Property description
script	string	

SuperNode Parameters

You can use scripts to create or set SuperNode parameters using the general format:

```
mySuperNode.setParameterValue("minvalue", 30)
```

You can retrieve the parameter value with:

```
value mySuperNode.getParameterValue("minvalue")
```

Finding Existing SuperNodes

You can find SuperNodes in streams using the `findByType()` function:

```
source_supernode = modeler.script.stream().findByType("source_super", None)
process_supernode = modeler.script.stream().findByType("process_super", None)
terminal_supernode = modeler.script.stream().findByType("terminal_super", None)
```

Setting Properties for Encapsulated Nodes

You can set properties for specific nodes encapsulated within a SuperNode by accessing the child diagram within the SuperNode. For example, suppose you have a source SuperNode with an encapsulated Variable File node to read in the data. You can pass the name of the file to read (specified using the `full_filename` property) by accessing the child diagram and finding the relevant node as follows:

```
childDiagram = source_supernode.getChildDiagram()
varfilename = childDiagram.findByType("variablefile", None)
varfilename.setPropertyValue("full_filename", "c:/mydata.txt")
```

Creating SuperNodes

If you want to create a SuperNode and its content from scratch, you can do that in a similar way by creating the SuperNode, accessing the child diagram, and creating the nodes you want. You must also ensure that the nodes within the SuperNode diagram are also linked to the input- and/or output connector nodes. For example, if you want to create a process SuperNode:

```
process_supernode = modeler.script.stream().createAt("process_super", "My SuperNode", 200, 200)
childDiagram = process_supernode.getChildDiagram()
filternode = childDiagram.createAt("filter", "My Filter", 100, 100)
childDiagram.linkFromInputConnector(filternode)
childDiagram.linkToOutputConnector(filternode)
```

General Classes

This provides general classes used by other parts of the API.

- [enum Objects](#)
- [ModelerException Objects](#)
- [VersionInfo Objects](#)

Related information

- [enum Objects](#)
- [ModelerException Objects](#)
- [VersionInfo Objects](#)

enum Objects

This class provides the basis for all enumerated classes in the Modeler API. Note that enum names are not case-sensitive.

```
e.equals(object) : boolean
```

`object` (`Object`): the other object.

Returns `True` if the supplied object is equal to this object.

`e.getName() : string`

Returns the name of the enumeration. Same as `toString()`.

`e.hashCode() : int`

Returns the hash code for this object.

`e.isUnknown() : boolean`

Returns `True` if this enumeration value represents an unknown or undefined object or `False` otherwise. An enumeration may contain at most one value that represents an unknown or undefined value.

`e.toString() : string`

Returns the name of the enumeration. Same as `getName()`.

Related information

- [General Classes](#)

ModelerException Objects

A generic Predictive Server exception.

Related information

- [General Classes](#)

VersionInfo Objects

Default Modeler API version information. The version information is made up of the major version, minor version and patch version. A change in the major version number indicates significant extensions to the functionality; a change in the minor version number indicates small extensions or refinements to the functionality; a change in the patch version number indicates a bug-fix release and that the documented functionality has not changed.

Constants:

`MAJOR_VERSION (int) :`
`MINOR_VERSION (int) :`
`RELEASE_VERSION (int) :`

`getBuildDate() : Date`

Returns the build date and time. If the build time is not available then `None` is returned.

`getBuildVersion() : string`

Returns the build version of this library. If the build version is not available then the empty string is returned.

`getCopyright() : string`

Returns the copyright message.

`getMajorVersion() : int`

Returns the major version number.

`getMinorVersion() : int`

Returns the minor version number.

`getName() : string`

Returns the name of the library.

`getPatchVersion() : int`

Returns the patch version number.

Related information

- [General Classes](#)
-

Content Management

This provides classes used for accessing content generated during stream execution.

- [ColumnStatsContentModel Objects](#)
 - [ContentModel Objects](#)
 - [JSONContentModel Objects](#)
 - [PairwiseStatsContentModel Objects](#)
 - [StatisticType Objects](#)
 - [TableContentModel Objects](#)
 - [XMLContentModel Objects](#)
-

Related information

- [ColumnStatsContentModel Objects](#)
 - [ContentModel Objects](#)
 - [JSONContentModel Objects](#)
 - [PairwiseStatsContentModel Objects](#)
 - [StatisticType Objects](#)
 - [TableContentModel Objects](#)
 - [XMLContentModel Objects](#)
-

ColumnStatsContentModel Objects

Subclass of [ContentModel](#).

An interface that provides a mechanism for accessing column/univariate statistics from nodes which can produce them.

```
c.getAvailableColumns() : List
```

Returns the names of the columns that can be queried.

```
c.getAvailableStatistics() : List
```

Returns the statistics which have been computed. Note that not all columns may have values for all available statistics.

```
c.getStatistic(column, statistic) : Number
```

column (string) : the column

statistic (StatisticType) : the statistic

Returns the value of the statistic for the specified.

Related information

- [Content Management](#)
-

ContentModel Objects

An interface that provides a mechanism for accessing the content of a [ContentContainer](#) in a structured way.

```
c.getContainerName() : string
```

Returns the name of the container that this content model is associated with.

```
c.reset()
```

Flushes any internal storage associated with this content model.

Related information

- [Content Management](#)

JSONContentModel Objects

Subclass of **ContentModel**.

An interface that provides a mechanism for accessing the JSON content of a **ContentContainer**. Content is accessed using lists of strings or integers. Strings are used to access named children in the JSON object while integers are used to access specific values in JSON arrays.

```
j.getChildValuesAt(path, artifact) : Map
```

path (**List**) : the path to the required object

artifact (**JSONArtifact**) : the starting artifact or **None**

Returns the child values of the specified path if the path leads to a JSON object or **None** otherwise. The keys in the table are strings while the associated value may be a literal string, integer, real or boolean, or a JSON artifact (either a JSON object or a JSON array).

Exceptions:

Exception : if there is an error parsing the JSON or when accessing specific content.

```
j.getChildrenAt(path, artifact) : List
```

path (**List**) : the path to the required object

artifact (**JSONArtifact**) : the starting artifact or **None**

Returns the list of objects at the specified path if the path leads to a JSON array or **None** otherwise. The returned values may be a literal string, integer, real or boolean, or a JSON artifact (either a JSON object or a JSON array).

Exceptions:

Exception : if there is an error parsing the JSON or when accessing specific content.

```
j.getJSONAsString() : string
```

Returns the JSON content as a string.

```
j.getObjectAt(path, artifact) : object
```

path (**List**) : the path to the required object

artifact (**JSONArtifact**) : the starting artifact or **None**

Returns the object at the specified path. The supplied root artifact may be **None** in which case the root of the content is used. The returned value may be a literal string, integer, real or boolean, or a JSON artifact (either a JSON object or a JSON array).

Exceptions:

Exception : if there is an error parsing the JSON or when accessing specific content

Related information

- [Content Management](#)

PairwiseStatsContentModel Objects

Subclass of **ContentModel**.

An interface that provides a mechanism for accessing correlations between columns.

```
p.getAvailablePrimaryColumns() : List
```

Returns the names of the primary columns that can be queried. If the statistics are comparing values within a column then only a single primary column will be returned.

```
p.getAvailablePrimaryValues() : List
```

Returns the values of the primary column that can be queried.

```
p.getAvailableSecondaryColumns() : List
```

Returns the names of the secondary columns that can be queried.

```
p.getAvailableStatistics() : List
```

Returns the statistics which have been computed. Note that not all column pairs may have values for all available statistics.

```
p.getStatistic(primaryColumn, secondaryColumn, statistic) : Number
```

primaryColumn (string) : the primary column

secondaryColumn (string) : the secondary column

statistic (StatisticType) : the statistic

Returns the value of the statistic for the specified pair of columns. Note that the method always returns null if the primary and secondary columns are the same.

```
p.getStatistic(primaryColumn, primaryValue, secondaryColumn, statistic) : Number
```

primaryColumn (string) : the primary column

primaryValue (object) : the primary column value

secondaryColumn (string) : the secondary column

statistic (StatisticType) : the statistic

Returns the value of the statistic for the specified value in the primary column associated with the secondary column. Note that the method always returns null if the primary and secondary columns are the same.

Related information

- [Content Management](#)
-

StatisticType Objects

Defines the statistics that can be computed either for a single column or for combinations of columns/column values.

Constants:

Count (StatisticType) : Represents the number of non-null values.

Covariance (StatisticType) : Represents the covariance between 2 columns.

FTest (StatisticType) : Represents the F-test.

Kurtosis (StatisticType) : Represents the kurtosis.

KurtosisStandardError (StatisticType) : Represents the standard error of the kurtosis.

Max (StatisticType) : Represents the maximum value.

Mean (StatisticType) : Represents the mean value.

MeanStandardError (StatisticType) : Represents the standard error of the mean value.

Median (StatisticType) : Represents the median (middle) value.

Min (StatisticType) : Represents the minimum value.

Mode (StatisticType) : Represents the modal (most common) value.

Pearson (StatisticType) : Represents the Pearson correlation between 2 columns.

Range (StatisticType) : Represents the range value.

```

Skewness (StatisticType) : Represents the skewness.

SkewnessStandardError (StatisticType) : Represents the standard error of the skewness.

StandardDeviation (StatisticType) : Represents the standard deviation.

StandardErrorOfMean (StatisticType) : Represents the standard error of the mean.

Sum (StatisticType) : Represents the sum of values.

TTest (StatisticType) : Represents the t-test.

UniqueCount (StatisticType) : Represents the number of distinct values.

ValidCount (StatisticType) : Represents the number of valid values.

Variance (StatisticType) : Represents the variance.

s.isPairwise() : boolean

Returns whether the statistic is only appropriate when comparing more than one value.

valueOf(name) : StatisticType

name (string) :

values() : StatisticType[]

```

Related information

- [Content Management](#)
-

TableContentModel Objects

Subclass of **ContentModel**.

An interface that provides a mechanism for accessing the tabular content of a **ContentContainer**. Values are arranged in rows and named columns where values in a particular column have the same storage type.

t.getColumnName(columnIndex) : int

Returns the number of columns in this content model..

t.getColumnName(columnIndex) : string

columnIndex (int) : the column index

Returns the name of a specified column in this content model. Returns the empty string if the column does not have a name. The column index must be in the range: `0 <= index < getRowCount ()`.

Exceptions:

IndexOutOfBoundsException: unless the column index is in range

t.getRowCount () : int

Returns the number of rows in this content model.

t.getStorageType(columnIndex) : StorageType

columnIndex (int) : the column index

Returns the storage type of objects in a specified column of this content model. The column index must be in the range: `0 <= index < getRowCount ()`.

Exceptions:

IndexOutOfBoundsException: unless the column index is in range

t.getValueAt(rowIndex, columnIndex) : object

rowIndex (int) : the row index

`columnIndex (int) : the column index`

Returns the value from a specified row and column in this content model. The row and column indexes must be in the range: `0 <= rowIndex < getRowCount() && 0 <= columnIndex < getColumnCount()`. The returned value will either be `None` or an instance of the class returned by `getColumnClass` for the same column index.

Exceptions:

`IndexOutOfBoundsException`: unless the column indexes are in range

Related information

- [Content Management](#)
-

XMLContentModel Objects

Subclass of `ContentModel`.

An interface that provides a mechanism for accessing the XML content of a ContentContainer. Content is accessed using XPath syntax without having to configure the additional parsers although callers can access the underlying XML text if they prefer to use a different mechanism.

`x.getBooleanValue(xpath) : boolean`

`xpath (string) : the XPath expression`

Returns the boolean result of evaluating the specified path expression.

Exceptions:

`Exception`:

`x.getNumericValue(xpath) : Number`

`xpath (string) : the XPath expression`

Returns the result of evaluating the path with return type of numeric. For example, count the number of elements that match the path expression.

Exceptions:

`Exception`:

`x.getStringValue(xpath, attribute) : string`

`xpath (string) : the XPath expression`

`attribute (string) : the name of the attribute to be returned or None if the XML node value is to be returned`

Evaluates the XPath expression to an XML node and returns either the value of the specified attribute in that node if the attribute name is a non-zero length string or the node value otherwise.

Exceptions:

`Exception`:

`x.getStringValues(xpath, attribute) : List`

`xpath (string) : the XPath expression`

`attribute (string) : the name of the attribute to be returned or None if the XML node value is to be returned`

Evaluates the XPath expression to a collection of XML nodes and returns a list containing either the value of the specified attribute in that node if the attribute name is a non-zero length string or the node value otherwise.

Exceptions:

`Exception`:

`x.getValuesList(xpath, attributes, includeValue) : List`

`xpath (string) : the XPath expression`

attributes (`List`) : the list of attribute names whose values are to be returned in each sublist

includeValue (`boolean`) : whether the XML node value should be included at the end of each sublist

Evaluates the XPath expression to obtain a list of XML nodes and returns a list of lists where each sublist contains the specified values from a particular node. The values of the named attributes are added to each sublist and the node value is added to the end of the sublist if the `includeValue` flag is set to `True`.

Exceptions:

Exception:

`x.getValuesMap(xpath, keyAttribute, attributes, includeValue) : Map`

`xpath (string)` : the XPath expression

`keyAttribute (string)` :

`attributes (List)` : the list of attribute names whose values are to be returned in each table entry list

`includeValue (boolean)` : whether the XML node value should be included at the end of each table entry list

Evaluates the XPath expression to obtain a list of XML nodes and returns a hash table of the specified attributes. The key for each entry is either the value of the specified key attribute in that node if the key attribute name is a non-zero length string or the node value otherwise. The value for each entry in the table is a list containing the specified values from the key node. The node value is added to the end of each list if the `includeValue` flag is set to `True`.

Exceptions:

Exception:

`x.getXMLAsString() : string`

Returns the XML as a string.

`x.isNamespaceAware() : boolean`

Returns `True` if the content model is aware of XML namespaces when parsing XML documents. The default value is `False`.

`x.setNamespaceAware(isAware)`

`isAware (boolean)` : whether the content model should be namespace aware

Sets whether the content model is aware of XML namespaces when parsing XML documents. Modifying this value can affect the syntax that should be used in XPath expressions. By default, XML content models are not namespace aware. When the value is changed, `reset()` is called to ensure that all internal storage is flushed to ensure any documents are re-parsed with the new setting.

Related information

- [Content Management](#)
-

Core Objects

This provides the base objects used by other parts of the API.

- [ASCredentialDescriptor Objects](#)
- [ApplicationData Objects](#)
- [ContentFormat Objects](#)
- [ContentContainer Objects](#)
- [ContentContainerProvider Objects](#)
- [ContentProvider Objects](#)
- [CredentialDescriptor Objects](#)
- [FileFormat Objects](#)
- [IncompatibleServerException Objects](#)
- [InvalidPropertyException Objects](#)
- [ModelFieldRole Objects](#)
- [ObjectCreationException Objects](#)
- [ObjectLockedException Objects](#)
- [OwnerException Objects](#)
- [ParameterDefinition Objects](#)
- [ParameterProvider Objects](#)

- [ParameterStorage Objects](#)
- [ParameterType Objects](#)
- [PropertiedObject Objects](#)
- [RepositoryConnectionDescriptor Objects](#)
- [RepositoryConnectionDescriptor2 Objects](#)
- [RepositoryConnectionDescriptor3 Objects](#)
- [ServerConnectionDescriptor Objects](#)
- [ServerConnectionException Objects](#)
- [ServerVersionInfo Objects](#)
- [StructureAttributeType Objects](#)
- [StructuredValue Objects](#)
- [SystemServerConnectionDescriptor Objects](#)

Related information

- [ASCredentialDescriptor Objects](#)
- [ApplicationData Objects](#)
- [ContentFormat Objects](#)
- [ContentProvider Objects](#)
- [FileFormat Objects](#)
- [IncompatibleServerException Objects](#)
- [InvalidPropertyException Objects](#)
- [ModelFieldRole Objects](#)
- [ObjectCreationException Objects](#)
- [ObjectLockedException Objects](#)
- [OwnerException Objects](#)
- [ParameterDefinition Objects](#)
- [ParameterProvider Objects](#)
- [ParameterStorage Objects](#)
- [ParameterType Objects](#)
- [PropertiedObject Objects](#)
- [RepositoryConnectionDescriptor Objects](#)
- [RepositoryConnectionDescriptor2 Objects](#)
- [RepositoryConnectionDescriptor3 Objects](#)
- [ServerConnectionDescriptor Objects](#)
- [ServerConnectionException Objects](#)
- [ServerVersionInfo Objects](#)
- [StructureAttributeType Objects](#)
- [StructuredValue Objects](#)
- [SystemServerConnectionDescriptor Objects](#)

ASCredentialDescriptor Objects

Define credential to log in the Analytic Server

```
a.getPassword() : string
```

Get password

```
a.getUserName() : string
```

Get user name

Related information

- [Core Objects](#)

ApplicationData Objects

Some features of the Modeler API include information such as the name and version of the application that built them. This supports the ability to brand the Modeler API. By changing this information held by the `SessionFactory`, an application can modify the apparent source of an attribute. Note however that this does not affect "internal" version information associated with some Modeler API features (e.g., persistence).

```
a.getApplicationName() : string
```

Returns the name of the application e.g., "Predictive Server API".

```
a.getApplicationVersion() : string
```

Returns the version of the application e.g., "16.0.0".

```
a.getCopyright() : string
```

Returns the application copyright information e.g., "Copyright (c) 2004-2013, IBM Corp.".

```
a.getVendorName() : string
```

Returns the name of the vendor e.g., "IBM Corp.".

Related information

- [Core Objects](#)

ContentFormat Objects

Subclass of [Enum](#).

Enumerates the valid content formats for content containers.

Constants:

```
BINARY (ContentFormat) : Indicates that the format is binary.
```

```
UTF8 (ContentFormat) : Indicates that the format is UTF8.
```

```
getEnum(name) : ContentFormat
```

```
name (string) : the enumeration name
```

Returns the enumeration with the supplied name or `None` if no enumeration exists for the supplied name.

```
getValues() : ContentFormat[]
```

Returns an array containing all the valid values for this enumeration class.

Related information

- [Core Objects](#)

ContentContainer Objects

Defines the functionality associated with a content container. A content container is used to hold objects (typically generated as a result of executing a stream) which are stored as part of a node or output object.

```
c.getContent() : object
```

Returns the content. The return value will either be a string if the content format is `ContentFormat.UTF8` or a byte array if the content format is `ContentFormat.BINARY`. An empty container will return `None`.

```
c.getContentAsBinary() : byte[]
```

Returns the binary content if the content format is `ContentFormat.BINARY` or `None` otherwise.

```
c.getContentAsUTF8() : string
```

Returns the UTF8 content if the content format is `ContentFormat.UTF8` or `None` otherwise.

```
c.getContentFormat() : ContentFormat
```

Returns the format used to store the component's content. An empty container will return `None`.

```
c.getName() : string
```

Returns the name of this container.

```
c.getTypeID() : string
```

Returns the ID of the container type declaration.

```
c.isEmpty() : boolean
```

Returns true if this container is empty; that is, it has no content associated with it.

```
c.readContainer(inputStream, contentFormat)
```

```
 inputStream (InputStream) :
```

```
contentFormat (ContentFormat) : the content format to be used for storing the input
```

Reads the container from the specified input stream. The input stream is read as a byte array. If the content format is `ContentFormat.BINARY` then the byte array is stored as-is. If the content format is `ContentFormat.UTF8` then the byte array is encoded as a UTF-8 string.

Exceptions:

```
java.io.IOException : if the implementing code throws an IOException.
```

```
c.writeContainer(outputStream)
```

```
 outputStream (OutputStream) : the output stream
```

Writes the container to the specified output stream.

Exceptions:

```
java.io.IOException : if the implementing code throws an IOException.
```

Related information

- [Core Objects](#)

ContentContainerProvider Objects

An interface that provides the basic mechanism for accessing and updating generated objects.

```
c.getContainerTagIterator() : Iterator
```

Returns an iterator for the container tags where each tag is a string.

```
c.getContainerTags() : List[]
```

Returns a list of the container tags.

```
c.getContainerTypeID(tag) : string
```

```
tag (string) : the container tag
```

Returns the type ID associated with the container tag or `None` if the container is unknown or has no type ID.

```
c.getContentContainer(tag) : ContentContainer
```

```
tag (string) : the container tag
```

Returns the container associated with the supplied tag or `None`.

```
c.getContentModel(tag) : ContentModel
```

```
tag (string) : the content model tag
```

Returns the content model associated with the content tags or `None` if content model tag is unknown.

```
c.getContentModelTags() : List
```

Returns a list of the content models supported by this object.

```
c.putContentContainer(tag, container)
```

```
tag (string) : the container tag
```

`container (ContentContainer)` : the container or `None`.

Adds or updates the container component with the supplied tag.

Related information

- [Core Objects](#)
-

ContentProvider Objects

Stores and retrieves content for an application. Content is identified using *tags* defined by the application. It is the application's responsibility to ensure that tags are unique, and a class-like naming scheme is recommended, e.g. `x.y.z....`

`c.getContent(tag) : Object`

`tag (string)` : the content tag

Returns the content identified by the specified tag or `None` if the tag is unknown to this provider. The type of the result is determined by the content format.

`c.getContentAsBinary(tag) : byte[]`

`tag (string)` : the content tag

Returns the binary content identified by the specified tag or `None` if the tag is unknown to this provider or the content does not have binary format.

`c.getContentAsUTF8(tag) : string`

`tag (string)` : the content tag

Returns the string content identified by the specified tag or `None` if the tag is unknown to this provider or the content does not have string format.

`c.getContentFormat(tag) : ContentFormat`

`tag (string)` : the content tag

Returns the format of the content identified by the specified tag or `None` if the tag is unknown to this provider. The format determines the content type.

`c.getContentTagIterator() : Iterator`

Returns an iterator over the content tags known to this provider.

`c.isContentCurrent(tag) : boolean`

`tag (string)` : the content tag

Returns `True` if the specified content is considered current (up-to-date) with respect to its context, e.g. its containing stream. Returns `False` if the content tag is unknown to this provider, or as a hint that the content may be stale.

`c.putContent(tag, format, content)`

`tag (string)` : the content tag

`format (ContentFormat)` : the content format

`content (Object)` : the content

Stores content in this provider with the specified tag and format. The content must not be `None` and its type must be compatible with the format. Any existing content with the same tag is replaced.

`c.putContentAsBinary(tag, content)`

`tag (string)` : the content tag

`content (byte[])` : the content

Stores binary content in this provider under the specified tag. The content must not be `None`.

`c.putContentAsUTF8(tag, content)`

`tag (string)` : the content tag

```
content (string) : the content
```

Stores string content in this provider under the specified tag. The content must not be **None**.

```
c.removeContent(tag)
```

```
tag (string) : the content tag
```

Removes any content stored by this provider under the specified tag.

Related information

- [Core Objects](#)

CredentialDescriptor Objects

This defines the credentials to log into DataBase.

```
c.getDefinitionName() : string
```

Returns the credential definition name.

```
c.getDomain() : string
```

Returns the domain associated with this credential.

```
c.getPassword() : string
```

Returns the password.

```
c.getUserName() : string
```

Returns the user name.

Related information

- [Core Objects](#)

FileFormat Objects

Subclass of **Enum**.

This class defines constants for the different file formats supported by objects in the Modeler API.

Constants:

```
BITMAP (FileFormat) : Identifier for bitmap image format.
```

```
COGNOS_ACTIVE_REPORT (FileFormat) : Identifier for Cognos Active Report format.
```

```
COMMA_DELIMITED (FileFormat) : Identifier for comma-delimited text.
```

```
COMPOSITE_PROCESSOR (FileFormat) : Identifier for the native SuperNode format.
```

```
DOCUMENT_OUTPUT (FileFormat) : Identifier for the native DocumentOutput format.
```

```
EMF (FileFormat) : Identifier for Enhanced Metafile format.
```

```
EPS (FileFormat) : Identifier for Encapsulated PostScript format.
```

```
EXTERNAL_MODULE_SPECIFICATION (FileFormat) : Identifier for the native external module specification format.
```

```
HTML (FileFormat) : Identifier for HTML format.
```

```
HTMLC (FileFormat) : Identifier for HTMLC format.
```

```
JAR (FileFormat) : Identifier for JAR (Java archive) format.
```

```
JPEG (FileFormat) : Identifier for JPEG image format.
```

MODEL_OUTPUT (**FileFormat**) : Identifier for the native **ModelOutput** format.

MODEL_OUTPUT_SET (**FileFormat**) : Identifier for the native **ModelOutput** set format.

MS_EXCEL (**FileFormat**) : Identifier for MS Excel format.

MS_EXCEL2007 (**FileFormat**) : Identifier for MS Excel 2007 format.

MS_EXCEL2007_M (**FileFormat**) : Identifier for MS Excel 2007 macro-enabled format.

MS_POWERPOINT (**FileFormat**) : Identifier for MS PowerPoint format.

MS_POWERPOINT2007 (**FileFormat**) : Identifier for MS PowerPoint 2007 format.

MS_WORD (**FileFormat**) : Identifier for MS Word format.

MS_WORD2007 (**FileFormat**) : Identifier for MS Word 2007 format.

PASW_RULE (**FileFormat**) : Identifier for PASW RULE format.

PDF (**FileFormat**) : Identifier for PDF format.

PLAIN_TEXT (**FileFormat**) : Identifier for plain text format.

PNG (**FileFormat**) : Identifier for PNG image format.

PROCESSOR (**FileFormat**) : Identifier for the native **Node** format.

PROCESSOR_STREAM (**FileFormat**) : Identifier for the native **Stream** format.

PROJECT (**FileFormat**) : Identifier for the native **Project** format.

PUBLISHED_IMAGE (**FileFormat**) : Identifier for the native image format produced by a Publisher node.

PUBLISHED_PARAMETERS (**FileFormat**) : Identifier for the native image parameter format produced by a Publisher node.

RTF (**FileFormat**) : Identifier for Rich Text Format.

SPSS_APPLICATION_VIEW (**FileFormat**) : Identifier for SPSS application view format.

SPSS_DATA (**FileFormat**) : Identifier for SPSS .sav data format.

SPSS_DATA_PROVIDER (**FileFormat**) : Identifier for SPSS data provider format.

SPSS_SCENARIO (**FileFormat**) : Identifier for SPSS scenario format.

SPSS_SCENARIO_TEMPLATE (**FileFormat**) : Identifier for SPSS scenario template format.

SPSS_WEB_REPORT (**FileFormat**) : Identifier for SPSS Web Report format.

SPV (**FileFormat**) : Identifier for SPV format.

SPW (**FileFormat**) : Identifier for SPW format.

STATE (**FileFormat**) : Identifier for the native combined **ModelOutput** and **Stream** format.

TAB_DELIMITED (**FileFormat**) : Identifier for tab-delimited text.

TIFF (**FileFormat**) : Identifier for TIFF format.

UNKNOWN (**FileFormat**) : Identifier for an unknown file format.

VIZ (**FileFormat**) : Identifier for VIZ format.

VIZML (**FileFormat**) : Identifier for VIZML format.

XML (**FileFormat**) : Identifier for XML format.

ZIP (**FileFormat**) : Identifier for Zip (compressed archive) format.

f.equals(object) : **boolean**

object (**Object**) :

Returns **True** if the supplied object is equal to this. Two file formats are considered equal if the names are equal ignoring the case of the name.

f.getDefaultExtension() : **string**

Returns the default file extension for this file format or an empty string if no file extensions are defined for this file format. The file extension includes the leading ":".

f.getEnum(name) : FileFormat

name (string) : the enumeration name

Returns the enumeration with the supplied name or **None** if no enumeration exists for the supplied name. The lookup is not case-sensitive.

f.getExtensionAt(index) : string

index (int) : the file extension index

Returns the file extension at the supplied index. The first (0'th) extension is assumed to be the default extension. The file extension includes the leading ":".

f.getExtensionCount() : int

Returns the number of file extensions associated with this file format.

f.getFileFilterPattern() : string

Returns a file filter pattern suitable for use with a file chooser dialog that includes all the file extensions associated with this file type. An empty string is returned if no file extensions are defined for this file type.

getFileFormatForExtension(extension) : FileFormat

extension (string) : the file extension

Returns the default file format for the supplied extension or **None** if no matching file format can be found. The extension search is not case-sensitive.

getFileFormatForMIMEType(mimeType) : FileFormat

mimeType (string) : the mime type to be searched for

Returns the file format for the supplied MIME type or **None** if no matching file format can be found.

getFileFormatsForExtension(extension) : List

extension (string) : the file extension

Returns a list of all file formats with the supplied extension. The extension search is not case-sensitive.

f.getMIMEType() : string

Returns the MIME type associated with this file format or an empty string if no MIME type has been defined.

getValues() : FileFormat[]

Returns an array containing all the valid values for this enumeration class.

f.isUnknown() : boolean

Returns **True** if this value is **UNKNOWN**.

f.isValidExtension(extension) : boolean

extension (string) : the file extension

Returns **True** if the supplied extension is valid for this file format or **False** otherwise. **None** values and the empty string always return **False**.

Related information

- [Core Objects](#)
-

IncompatibleServerException Objects

Subclass of **ServerConnectionException**.

An exception thrown while attempting to connect to an incompatible server version.

A message string is always created for this exception.

i.getServerMajorVersion() : int

Returns the major version returned by the server being connected to.

```
i.getServerMinorVersion() : int
```

Returns the minor version returned by the server being connected to.

```
i.getServerPatchVersion() : int
```

Returns the patch version returned by the server being connected to.

Related information

- [Core Objects](#)

InvalidPropertyException Objects

Subclass of [ModelerException](#).

An exception thrown when an invalid property is accessed or an invalid value is supplied for a property.

A message string is always created for this exception.

```
i.getProperty() : string
```

Returns the property name.

```
i.getValue() : Object
```

Returns the value being assigned or `None`.

Related information

- [Core Objects](#)

ModelFieldRole Objects

Subclass of [Enum](#).

This class enumerates the valid model output roles.

Constants:

`ADJUSTED_PROBABILITY (ModelFieldRole)` : Indicates a field containing the adjusted probability of the target value.

`ENTITY_AFFINITY (ModelFieldRole)` : Indicates a field containing the affinity or distance of the value from the entity.

`ENTITY_ID (ModelFieldRole)` : Indicates a field containing the entity ID associated with the cluster, tree node, neuron or rule.

`LOWER_CONFIDENCE_LIMIT (ModelFieldRole)` : Indicates a field containing the lower confidence limit of a numeric prediction.

`PREDICTED_DISPLAY_VALUE (ModelFieldRole)` : Indicates a field containing a displayed-friendly representation of the predicted value.

`PREDICTED_VALUE (ModelFieldRole)` : Indicates a field containing the predicted value.

`PROBABILITY (ModelFieldRole)` : Indicates a field containing the probability of the target value.

`PREDICTED_SEQUENCE (ModelFieldRole)` : Indicates a field containing the predicted sequence.

`PREDICTED_SEQUENCE_PROBABILITY (ModelFieldRole)` : Indicates a field containing the probability associated with the predicted sequence.

`PROPENSITY (ModelFieldRole)` : Indicates a field containing the propensity of the prediction.

`RESIDUAL (ModelFieldRole)` : Indicates a field containing the residual of the predicted numeric value.

`STANDARD_DEVIATION (ModelFieldRole)` : Indicates a field containing the standard deviation of the target value.

`STANDARD_ERROR (ModelFieldRole)` : Indicates a field containing the standard error of the predicted numeric value.

SUPPLEMENTARY (`ModelFieldRole`) : Indicates a field containing some additional information about the value that cannot be categorized into one of the other model field roles.

SUPPORT (`ModelFieldRole`) : Indicates a field containing the support associated with an association model rule.

UNKNOWN (`ModelFieldRole`) : Indicates that the role cannot be determined.

UPPER_CONFIDENCE_LIMIT (`ModelFieldRole`) : Indicates a field containing the upper confidence limit of a numeric prediction.

VALUE (`ModelFieldRole`) : Indicates a field containing the model output value for non-predictive models.

VARIANCE (`ModelFieldRole`) : Indicates a field containing the variance of the target value.

`getEnum(name) : ModelFieldRole`

`name (string) : the enumeration name`

Returns the enumeration with the supplied name or None if no enumeration exists for the supplied name.

`getValues() : ModelFieldRole[]`

Returns an array containing all the valid values for this enumeration class.

`m.isUnknown() : boolean`

Returns True if this value is UNKNOWN.

Related information

- [Core Objects](#)

ObjectCreationException Objects

Subclass of `ModelerException`.

An exception thrown when an instance of an object cannot be created.

No message string is set for this exception.

Related information

- [Core Objects](#)

ObjectLockedException Objects

Subclass of `ModelerException`.

An exception thrown when an attempt is made to lock or modify an object that is already locked. Locking an object prevents an updates to that object while the lock is in effect. A `Stream` and all its `Nodes` are locked by the `Session` during execution.

No message string is set for this exception.

`o.getObject() : Object`

Returns the object.

Related information

- [Core Objects](#)

OwnerException Objects

Subclass of `ModelerException`.

An exception thrown when an object owner requires an owned object.

No message string is set for this exception.

`o.getOwner() : Object`

Returns the object owner.

`o.getUnownedObject() : Object`

Returns the object that is not owned by the owner.

Related information

- [Core Objects](#)
-

ParameterDefinition Objects

Describes a parameter which affects the behaviour of some Modeler API objects. A parameter definition is obtained from a **ParameterProvider**. The parameter name is unique within the provider.

Parameters are modified through the provider. A **ParameterDefinition** instance may be a snapshot of a parameter definition at the time it was obtained from the provider and need not reflect subsequent modifications.

`p.getFalseFlag() : Object`

Returns the *false* indicator for this flag parameter. Returns `None` if the parameter type is other than **ParameterType.FLAG** or if no false value has been declared for the flag.

`p.getLowerBound() : Object`

Returns a lower bound on the valid values of this parameter. Returns `None` if the parameter type is other than **ParameterType.RANGE** or if no lower bound has been declared for the range.

`p.getParameterLabel() : string`

Returns a label for this parameter. The default label is the empty string.

`p.getParameterName() : string`

Returns the name of this parameter. The name is unique within the parameter provider.

`p.getParameterStorage() : ParameterStorage`

Returns the storage type of this parameter.

`p.getParameterType() : ParameterType`

Returns the measure of this parameter.

`p.getParameterValue() : Object`

Returns the value of this parameter.

`p.getSetValues() : Object[]`

Returns the valid values of this parameter. Returns `None` if the parameter type is other than **ParameterType.SET** or if no values have been declared for the set.

`p.getTrueFlag() : Object`

Returns the *true* indicator for this flag parameter. Returns `None` if the parameter type is other than **ParameterType.FLAG** or if no true value has been declared for the flag.

`p.getUpperBound() : Object`

Returns an upper bound on the valid values of this parameter. Returns `None` if the parameter type is other than **ParameterType.RANGE** or if no upper bound has been declared for the range.

`p.isValidValue(value) : boolean`

`value (Object) : the value`

Returns `True` if the specified value is valid for this parameter. The value must be compatible with the parameter storage type and with the parameter values when they are specified.

Related information

- [Core Objects](#)
-

ParameterProvider Objects

Identifies objects that contain parameters. Parameters do not control the behaviour of the implementing class directly but provide a look-up mechanism for values that may affect the behaviour of other objects.

`p.getParameterDefinition(parameterName) : ParameterDefinition`

`parameterName (string) : the parameter name`

Returns the parameter definition for the parameter with the specified name or `None` if no such parameter exists in this provider. The result may be a snapshot of the definition at the time the method was called and need not reflect any subsequent modifications made to the parameter through this provider.

`p.getParameterLabel(parameterName) : string`

`parameterName (string) : the parameter name`

Returns the label of the named parameter or `None` if no such parameter exists.

`p.getParameterStorage(parameterName) : ParameterStorage`

`parameterName (string) : the parameter name`

Returns the storage of the named parameter or `None` if no such parameter exists.

`p.getParameterType(parameterName) : ParameterType`

`parameterName (string) : the parameter name`

Returns the type of the named parameter or `None` if no such parameter exists.

`p.getParameterValue(parameterName) : Object`

`parameterName (string) : the parameter name`

Returns the value of the named parameter or `None` if no such parameter exists.

`p.parameterIterator() : Iterator`

Returns an iterator of parameter names for this object.

`p.setParameterLabel(parameterName, label)`

`parameterName (string) : the parameter name`

`label (string) : the parameter label`

Sets the label of the named parameter.

Exceptions:

`ObjectLockedException` : if the parameter provider is locked

`p.setParameterStorage(parameterName, storage)`

`parameterName (string) : the parameter name`

`storage (ParameterStorage) : the parameter storage`

Sets the storage of the named parameter.

Exceptions:

`ObjectLockedException` : if the parameter provider is locked

`p.setParameterType(parameterName, type)`

`parameterName (string) : the parameter name`

`type (ParameterType) : the parameter type`

Sets the type of the named parameter.

Exceptions:

`ObjectLockedException` : if the parameter provider is locked

`p.setParameterValue(parameterName, value)`

`parameterName (string)` : the parameter name

`value (Object)` : the parameter value

Sets the value of the named parameter.

Exceptions:

`ObjectLockedException` : if the parameter provider is locked

Related information

- [Core Objects](#)

ParameterStorage Objects

Subclass of `Enum`.

This class enumerates the valid storage types for columns in a `DataModel` or parameters.

Constants:

`DATE (ParameterStorage)` : Indicates that the storage type is a date type.

`INTEGER (ParameterStorage)` : Indicates that the storage type is an integer.

`REAL (ParameterStorage)` : Indicates that the storage type is a floating point number.

`STRING (ParameterStorage)` : Indicates that the storage type is a string.

`TIME (ParameterStorage)` : Indicates that the storage type is a time type.

`TIMESTAMP (ParameterStorage)` : Indicates that the storage type is a combined time and date type.

`UNKNOWN (ParameterStorage)` : Indicates that the storage type is unknown.

`getEnum(name) : ParameterStorage`

`name (string)` : the enumeration name

Returns the enumeration with the supplied name or `None` if no enumeration exists for the supplied name.

`p.getStorageClass() : Class`

Returns the Java class that parameter values with this storage will be returned as.

`getValues() : ParameterStorage[]`

Returns an array containing all the valid values for this enumeration class.

Related information

- [Core Objects](#)

ParameterType Objects

Subclass of `Enum`.

This class enumerates the valid types of parameter support by a `ParameterProvider`.

Constants:

FLAG (ParameterType) : Indicates that the parameter value is one of two values.

RANGE (ParameterType) : Indicates that the parameter value lies between two points on a scale.

SET (ParameterType) : Indicates that the parameter value is one of a (small) set of discrete values.

TYPELESS (ParameterType) : Indicates that the parameter can have any value compatible with its storage.

getEnum(name) : ParameterType

name (string) : the enumeration name

Returns the enumeration with the supplied name or **None** if no enumeration exists for the supplied name.

getValues() : ParameterType[]

Returns an array containing all the valid values for this enumeration class.

Related information

- [Core Objects](#)

PropertyiedObject Objects

This encapsulates the functionality of objects that contain settable properties. Properties are split into two sub-groups:

- **Simple properties:** these are basic name/value pairs
- **Keyed properties:** these are properties associated with specific objects

An example of a simple property is a Boolean value that specifies to a data file reader whether column names are included in the first line of a file:

```
myDataReader.setPropertyValue("read_field_names", Boolean.TRUE);
```

An example of a keyed property is used in an object that allows fields or columns to be removed from the data model. Here the setting is made up of both the property name and the key which is the column name:

```
// Remove "DateOfBirth" column  
myColumnFilter.setKeyedPropertyValue("include", "DateOfBirth", Boolean.FALSE);
```

Finally note that properties which correspond with **Enum** values must be converted to strings by calling `getName()` before being passed to one of the setter methods:

```
typeProcessor.setKeyedPropertyValue("role", "Drug", ModelingRole.OUT.getName());
```

Similarly, the getter methods will return Strings which can then be converted to the appropriate **Enum** value:

```
String value = (String) typeProcessor.getKeyedPropertyValue("role", "Drug");  
ModelingRole role = ModelingRole.getValue(value);
```

p.getKeyedPropertyKeys(propertyName) : List

propertyName (string) : the property name

Returns the keys currently defined for the supplied keyed property name.

p.getKeyedPropertyValue(propertyName, keyName) : Object

propertyName (string) : the property name

keyName (string) : the key name

Returns the value of the named property and key or **None** if no such property or key exists.

p.getLabel() : string

Returns the object's display label. The label is the value of the property "`custom_name`" if that is a non-empty string and the "`use_custom_name`" property is not set; otherwise, it is the value of `getName()`.

p.getName() : string

Returns the object's name.

p.getPropertyValue(propertyName) : Object

`propertyName (string) : the property name`

Returns the value of the named property or None if no such property exists.

`p.getSavedByVersion() : double`

Return the object's saved by version.

Exceptions:

`Exception` : if the information cannot be accessed for some reason

`p.isKeyedProperty(propertyName) : boolean`

`propertyName (string) : the property name`

Returns True if the supplied property name is a keyed property.

`p.isServerConnectionRequiredProperty(propertyName) : boolean`

`propertyName (string) : the property name`

Returns True if the supplied property name should only be set if there is a valid server connection.

`p.propertyIterator() : Iterator`

Returns an iterator of property names for this object.

`p.setKeyedPropertyValue(propertyName, keyName, value)`

`propertyName (string) : the property name`

`keyName (string) : the key name`

`value (Object) : the property value`

Sets the value of the named property and key.

Exceptions:

`ObjectLockedException` : if the propertied object is locked

`InvalidPropertyException` : if the property name is unknown or the supplied value is not valid for the property

`p.setLabel(label)`

`label (string) : the object's display label`

Sets the object's display label. If the new label is a non-empty string it is assigned to the property "custom_name", and False is assigned to the property "use_custom_name" so that the specified label takes precedence; otherwise, an empty string is assigned to the property "custom_name", and True is assigned to the property .

`p.setPropertyValue(propertyName, value)`

`propertyName (string) : the property name`

`value (Object) : the property value`

Sets the value of the named property.

Exceptions:

`ObjectLockedException` : if the propertied object is locked

`InvalidPropertyException` : if the property name is unknown or the supplied value is not valid for the property

`p.setPropertyValues(properties)`

`properties (Map) : the property/value pairs to be assigned`

Sets the values of the named properties. Each entry in the Map consists of a key representing the property name and the value which should be assigned to that property.

Exceptions:

`ObjectLockedException` : if the propertied object is locked

`InvalidPropertyException` : if one of the property names is unknown or one the supplied values is not valid for that property

`p.setPropertyValuesFrom(otherObject)`

`otherObject (PropertiedObject)` : the other object whose properties are to be copied

Sets the values of the properties that are in both this object and the supplied other object.

Exceptions:

`ObjectLockedException` : if the propertied object is locked

Related information

- [Core Objects](#)

RepositoryConnectionDescriptor Objects

Defines the basic data elements required to connect to a content repository.

`r.getDomainName() : string`

Returns the domain name to be used when logging into the content repository (if applicable) or the empty string.

`r.getHostNames() : string`

Returns the host name or IP address of the content repository machine.

`r.getPassword() : string`

Returns the user's password in plain text. An empty string indicates that there is no password.

`r.getPortNumber() : int`

Returns the port number which the content repository is listening on.

`r.getUseSSL() : boolean`

Returns `True` if the connection should use a secure socket.

`r.getUserName() : string`

Returns the user's login name.

Related information

- [Core Objects](#)

RepositoryConnectionDescriptor2 Objects

Subclass of `RepositoryConnectionDescriptor`.

Extends the repository connection details for applications which are aware of PASW platform services.

`r.getSsoToken() : Object`

Returns the SSO security token or `None` if there is no token.

`r.getSubject() : Object`

Returns the platform security subject or `None` if there is no subject.

Related information

- [Core Objects](#)

RepositoryConnectionDescriptor3 Objects

Subclass of `RepositoryConnectionDescriptor2`.

```
r.getContextRoot() : string
```

Return the context root of the content repository service.

Related information

- [Core Objects](#)

ServerConnectionDescriptor Objects

Subclass of **SystemServerConnectionDescriptor**.

Defines the basic data elements required to connect to a remote server.

```
s.getPassword() : string
```

Returns the user's password in plain text. An empty string indicates that there is no password.

```
s.isSSOConnect() : boolean
```

Returns `True` if the connection should use SSO.

```
s.isUseSSL() : boolean
```

Returns `True` if the connection should use SSL.

Related information

- [Core Objects](#)

ServerConnectionException Objects

Subclass of **ModelerException**.

An exception thrown while connecting or using a **ServerConnectionDescriptor**.

A message string is always created for this exception.

Related information

- [Core Objects](#)

ServerVersionInfo Objects

Version information for a connected server.

A server version has four components, in decreasing order of priority:

- major
- minor
- release
- fix pack

```
s.getServerVersionFixPack() : int
```

Returns the server fix pack number.

```
s.getServerVersionMajor() : int
```

Returns the server major version number.

```
s.getServerVersionMinor() : int
```

Returns the server minor version number.

```
s.getServerVersionRelease() : int
```

Returns the server release number.

Related information

- [Core Objects](#)

StructureAttributeType Objects

Subclass of **Enum**.

This class enumerates the valid types of attribute supported by a **ParameterProvider**.

Constants:

```
BOOLEAN (StructureAttributeType) :
```

```
DATE (StructureAttributeType) :
```

```
DOUBLE (StructureAttributeType) :
```

```
INTEGER (StructureAttributeType) :
```

```
STRING (StructureAttributeType) :
```

```
getEnum(name) : StructureAttributeType
```

```
name (string) : the enumeration name
```

Returns the enumeration with the supplied name or **None** if no enumeration exists for the supplied name.

```
s.getStorageClass() : Class
```

Returns the basic Java class used to represent this attribute's value. One of:

- **string**
- **int**
- **float**
- **boolean**
- **Date**

```
getValues() : StructureAttributeType[]
```

Returns an array containing all the valid values for this enumeration class.

Related information

- [Core Objects](#)

StructuredValue Objects

This interface defines a structured value. A structured value consists of a number of simple attributes.

```
s.changeAttributeValue(index, value) : StructuredValue
```

```
index (int) : the index
```

```
value (Object) : the new value
```

Returns a new structured value with the attribute value at the specified index changed to the new value. An attribute value cannot be **None** -- if the supplied value is **None**, it will be ignored.

```
s.getAttributeCount() : int
```

Returns the number of attribute values in the structure.

```
s.getAttributeValue(index) : Object
```

```
index (int) : the index
```

Returns the attribute value at the specified index.

Related information

- [Core Objects](#)

SystemServerConnectionDescriptor Objects

Defines the basic data elements that describes a server connection. Note that for security reasons, the user's password is not accessible through this interface.

`s.getDataDirectory() : string`

Returns the data directory for this connection descriptor.

`s.getDomainName() : string`

Returns the domain name to be used when logging into the server (if applicable) or the empty string.

`s.getHostNames() : string`

Returns the host name or IP address of the server machine.

`s.getPortNumber() : int`

Returns the port number which the server is listening on.

`s.getServerMajorVersion() : int`

Returns the major version of server required by this descriptor.

`s.getServerMinorVersion() : int`

Returns the minor version of server required by this descriptor.

`s.getServerPatchVersion() : int`

Returns the patch version of server required by this descriptor.

`s.getUserName() : string`

Returns the user's login name.

Related information

- [Core Objects](#)

Data and Metadata

This provides support for metadata such as the names, values and types of data.

- [Column Objects](#)
- [ColumnCountException Objects](#)
- [ColumnGroup Objects](#)
- [ColumnGroupType Objects](#)
- [DataModel Objects](#)
- [DataModelException Objects](#)
- [DataModelFactory Objects](#)
- [ExtendedMeasure Objects](#)
- [ExtendedStorage Objects](#)
- [GeoType Objects](#)
- [GeometryType Objects](#)
- [GlobalValues Objects](#)
- [GlobalValues.Type Objects](#)
- [InvalidColumnExceptionValues Objects](#)
- [ListStorage Objects](#)
- [MeasureType Objects](#)

- [MissingValueDefinition Objects](#)
- [ModelOutputMetadata Objects](#)
- [ModelingRole Objects](#)
- [RowSet Objects](#)
- [StorageType Objects](#)
- [UnknownColumnException Objects](#)

Related information

- [Column Objects](#)
- [ColumnCountException Objects](#)
- [ColumnGroup Objects](#)
- [ColumnGroupType Objects](#)
- [DataModel Objects](#)
- [DataModelError Objects](#)
- [GlobalValues Objects](#)
- [GlobalValues.Type Objects](#)
- [InvalidColumnExceptionValues Objects](#)
- [MeasureType Objects](#)
- [MissingValueDefinition Objects](#)
- [ModelOutputMetadata Objects](#)
- [ModelingRole Objects](#)
- [RowSet Objects](#)
- [StorageType Objects](#)
- [UnknownColumnException Objects](#)

Column Objects

This defines the properties of a set of columns. Applications that wish to construct `Column` instances must do so using the `DataModelFactory` rather than implementing the interface directly.

`c.getColumnLabel() : string`

Returns the label of the column or an empty string if there is no label associated with the column.

`c.getColumnName() : string`

Returns the name of the column.

`c.getFalseFlag() : Object`

Returns the "false" indicator value for the column, or `None` if either the value is not known or the column is not a flag.

`c.getLowerBound() : Object`

Returns the lower bound value for the values in the column, or `None` if either the value is not known or the column is not continuous.

`c.getMeasureType() : MeasureType`

Returns the measure type for the column.

`c.getMissingValueDefinition() : MissingValueDefinition`

Returns the missing value definition for the column or `None`.

`c.getModelOutputMetadata() : ModelOutputMetadata`

Returns the model output column role for the column if it is a model output column or `None`.

`c.getModelingRole() : ModelingRole`

Returns the modeling role for the column.

`c.getSetValues() : Object[]`

Returns an array of valid values for the column, or `None` if either the values are not known or the column is not a set.

`c.getStorageType() : StorageType`

Returns the storage type for the column.

`c.getTrueFlag() : Object`

Returns the "true" indicator value for the column, or `None` if either the value is not known or the column is not a flag.

`c.getUpperBound() : Object`

Returns the upper bound value for the values in the column, or `None` if either the value is not known or the column is not continuous.

`c.getValueLabel(value) : string`

`value (Object) : the value`

Returns the label for the value in the column or an empty string if there is no label associated with the value.

`c.hasMissingValueDefinition() : boolean`

Returns `True` if the column has a missing value definition.

`c.isMeasureDiscrete() : boolean`

Returns `True` if the column is discrete. Columns that are either a set or a flag are considered discrete.

`c.isModelOutputColumn() : boolean`

Returns `True` if this is a model output column.

`c.isStorageDatetime() : boolean`

Returns `True` if the column's storage is a time, date or timestamp value.

`c.isStorageNumeric() : boolean`

Returns `True` if the column's storage is an integer or a real number.

`c.isValidValue(value) : boolean`

`value (Object) : the value`

Returns `True` if the specified value is valid for this storage and valid when the valid column values are known.

`d.getExtendedMeasure(name) : ExtendedMeasure`

`name (string) : the column name`

Returns additional metadata that is specific for this field. For example, geospatial fields may return a geospatial-specific descriptor that defines the metadata necessary for the correct interpretation of the values.

Exceptions:

`DataModelError : if the named column does not exist`

`d.getExtendedStorage(name) : ExtendedStorage`

`name (string) : the field name`

Returns additional metadata for this field's storage or `None` if no additional storage metadata is defined.

`d.isList(name) : boolean`

`name (string) :`

Returns `True` if this field is a list field.

Related information

- [Data and Metadata](#)
-

ColumnCountException Objects

Subclass of `DataModelError`.

A `ColumnCountException` is thrown when the number of columns in the data did not match the number of columns specified in the data model.

No message string is set for this exception.

`c.getValues() : List`

Returns the list of values that were being converted.

Related information

- [Data and Metadata](#)

ColumnGroup Objects

This defines the properties of a column group. A column group is typically used to identify related columns e.g. that represent a multi response set.

```
c.getColumnGroupLabel() : string
```

Returns the label of the column group or an empty string if there is no label associated with the column group.

```
c.getColumnGroupName() : string
```

Returns the name of the column group.

```
c.getColumnGroupType() : ColumnGroupType
```

Returns the type of column group.

```
c.getColumnNames() : List
```

Returns the list of column names in this group. The list is a copy of the list in the group and can be modified without affecting the group definition.

```
c.getCountedValue() : string
```

Returns the counted value for this column group. The counted value only has meaning for multi dichotomy sets and represents the "true" or "selected" value of the column. Note that the value is represented as a string even if the columns named by the column group have storage other than string and it is the responsibility of the calling application to convert the string representation into a valid value.

Related information

- [Data and Metadata](#)

ColumnGroupType Objects

Subclass of [Enum](#).

This class enumerates the valid types for a ColumnGroup.

Constants:

- **GENERAL** (`ColumnGroupType`) : Indicates that the there are no special semantics associated with this column group.
- **MULTI_CATEGORY_SET** (`ColumnGroupType`) : Indicates that this column group represents a multi category set.
- **MULTI_DICHOTOMY_SET** (`ColumnGroupType`) : Indicates that this column group represents a multi dichotomy set.
- **SPLIT_GROUP** (`ColumnGroupType`) : A group representing the split fields. Ideally the application will keep the fields in the split group consistent with the fields whose role is "split". However, components will use the group to determine the order for the split fields and will typically ignore the "split" role. There should be at most one split group in the data model.

```
getEnum(name) : ColumnGroupType
```

```
name (string) : the enumeration name
```

Returns the enumeration with the supplied name or None if no enumeration exists for the supplied name.

```
getValues() : ColumnGroupType[]
```

Returns an array containing all the valid values for this enumeration class.

Related information

- [Data and Metadata](#)

DataModel Objects

This defines the properties of a set of columns. Applications that wish to construct data model instances must do so using the **DataModelFactory** rather than implementing the interface directly.

d.columnGroupIterator() : Iterator

Returns an iterator that returns each column group in turn.

d.columnIterator() : Iterator

Returns an iterator that returns each column in the "natural" insert order. The iterator returns instances of Column.

d.contains(name) : boolean

name (string) : the column name

Returns **True** if a column with the supplied name exists in this **DataModel**, **False** otherwise.

d.getColumn(name) : Column

name (string) :

Returns the column with the specified name.

Exceptions:

DataModelError : if the named column does not exist

d.getColumnCount() : int

Returns the number of columns in this set.

d.getColumnGroup(name) : ColumnGroup

name (string) :

Returns the named column group or **None** if no such column group exists.

d.getColumnGroupCount() : int

Returns the number of column groups in this data model.

d.getColumnLabel(name) : string

name (string) : the column name

Returns the label of the named column or an empty string if there is no label associated with the column.

Exceptions:

DataModelError : if the named column does not exist

d.getFalseFlag(name) : Object

name (string) : the column name

Returns the "false" indicator value for the column, or **None** if either the value is not known or the column is not a flag.

Exceptions:

DataModelError : if the named column does not exist

d.getLowerBound(name) : Object

name (string) : the column name

Returns the lower bound value for the values in the named column, or **None** if either the value is not known or the column is not continuous.

Exceptions:

DataModelError : if the named column does not exist

d.getMeasureType(name) : MeasureType

name (string) : the column name

Returns the measure type for values in the named column.

Exceptions:

DataModelError : if the named column does not exist

d.getMissingValueDefinition(name) : MissingValueDefinition

name (string) : the column name

Returns the missing value definition for the column or **None**.

Exceptions:

DataModelError : if the named column does not exist

d.getModelingRole(name) : ModelingRole

name (string) : the column name

Returns the modeling role for the named column.

Exceptions:

DataModelError : if the named column does not exist

d.getSetValues(name) : Object[]

name (string) : the column name

Returns an array of valid values for the column, or **None** if either the values are not known or the column is not a set.

Exceptions:

DataModelError : if the named column does not exist

d.getStorageType(name) : StorageType

name (string) : the column name

Returns the storage type for values in the named column.

Exceptions:

DataModelError : if the named column does not exist

d.isTrueFlag(name) : Object

name (string) : the column name

Returns the "true" indicator value for the column, or **None** if either the value is not known or the column is not a flag.

Exceptions:

DataModelError : if the named column does not exist

d.getUpperBound(name) : Object

name (string) : the column name

Returns the upper bound value for the values in the named column, or **None** if either the value is not known or the column is not continuous.

Exceptions:

DataModelError : if the named column does not exist

d.getValueLabel(name, value) : string

name (string) : the column name

value (Object) : the value

Returns the label for the value in the named column or an empty string if there is no label associated with the value.

Exceptions:

DataModelError : if the named column does not exist

d.hasMissingValueDefinition(name) : boolean

name (string) : the column name

Returns **True** if the column has a missing value definition.

Exceptions:

DataModelError : if the named column does not exist

d.isMeasureDiscrete(name) : boolean

name (string) : the column name

Returns **True** if the column is discrete. Columns that are either a set or a flag are considered discrete.

Exceptions:

DataModelError : if the named column does not exist

d.isModelOutputColumn(name) : boolean

name (string) : the column name

Returns **True** if this is a model output column.

Exceptions:

DataModelError : if the named column does not exist

d.isStorageDatetime(name) : boolean

name (string) : the column name

Returns **True** if the column's storage is a time, date or timestamp value.

Exceptions:

DataModelError : if the named column does not exist

d.isStorageNumeric(name) : boolean

name (string) : the column name

Returns **True** if the column's storage is an integer or a real number.

Exceptions:

DataModelError : if the named column does not exist

d.isValidValue(name, value) : boolean

name (string) : the column name

value (Object) : the value

Returns **True** if the specified value is valid for this storage and valid when the valid column values are known.

Exceptions:

DataModelError : if the named column does not exist

d.nameIterator() : Iterator

Returns an iterator that returns the name of each column in the "natural" insert order.

d.toArray() : Column[]

Returns the data model as an array of columns. The columns are ordered in their natural/insert order.

Related information

- [Data and Metadata](#)
-

DataModelError Objects

Subclass of **ModelerException**.

An exception thrown when there is a mismatch between physical set of data and its expected data model.

No message string is set for this exception.

```
d.getDataModel() : DataModel
```

Returns the data model that was being used to convert the values.

Related information

- [Data and Metadata](#)

DataModelFactory Objects

A factory class that creates and manipulates instances of `Column` and `DataModel`.

```
d.createCollectionType(valueMeasure) : CollectionType
```

`valueMeasure (MeasureType)` : the measure that should be applied to each value in the list.

Creates a new `CollectionType` object. The definition specifies the measurement associated with individual values in the list. The storage cannot be either `MeasureType.COLLECTION` or `MeasureType.GEOSPATIAL`, and should also be consistent with the storage type associated with the simple values in the list column that this metadata is to be associated with.

```
d.createColumn(name, label, storageType, measureType, modelingRole) : Column
```

`name (string)` : the column name

`label (string)` : the column label (may be `None`)

`storageType (StorageType)` : the storage definition

`measureType (MeasureType)` : the measure definition (may be `None`)

`modelingRole (ModelingRole)` : the model role definition for each column (may be `None`)

Returns a `Column` with the supplied attributes.

```
d.createColumn(name, label, extendedStorage) : Column
```

`name (string)` : the column name

`label (string)` : the column label (may be `None`)

`extendedStorage (ExtendedStorage)` : the extended storage definition

Returns a `Column` with the supplied attributes. The measure type is set to `MeasureType.TYPELESS`.

```
d.createColumn(name, label, extendedStorage, extendedMeasure) : Column
```

`name (string)` : the column name

`label (string)` : the column label (may be `None`)

`extendedStorage (ExtendedStorage)` : the extended storage definition

`extendedMeasure (ExtendedMeasure)` : the extended measure definition

Returns a `Column` with the supplied attributes. The measure type is set to `MeasureType.TYPELESS`.

```
d.createColumn(name, label, sourceColumn) : Column
```

`name (string)` : the column name

`label (string)` : the column label (may be `None`)

`sourceColumn (Column)` : the source column

Returns a `Column` with the specified name and labels but with all other attributes the same as source column.

Exceptions:

`DataModelError` : if the source column was not created by the `DataModelFactory`.

```
d.createDataModel() : DataModel
```

Returns an empty `DataModel`.

```
d.createFlagColumn(name, label, storageType, falseValue, trueValue) : Column
```

name (string) : the column name

label (string) : the column label (may be **None**)

storageType (StorageType) : the storage definition

falseValue (object) : the false value

trueValue (object) : the true value

Returns a **Column** with the supplied attributes.

```
d.createGeoType(geometryType, coordinates, wellKnownID, coordinateSystemName) : GeoType
```

geometryType (GeometryType) : the type of geometry object

coordinates (int) : a positive integer representing the number

WellKnownID (int) : the "well-known ID" of the coordinate system or 0 if this is not known

coordinateSystemName (string) : the name of the coordinate system or the empty string if this is not known.

Creates a new **GeoType** object. The definition specifies the type of geometry object, the number of coordinates (typically 2 or 3) required to represent the geometry object, and the "well-known ID" and/or the coordinate system name.

```
d.createListStorage(depth, valueType) : ListStorage
```

depth (int) : how deeply nested the simple values are in the list

valueType (StorageType) : the type of the simple values

Creates a new **ListStorage** object. The definition specifies the depth of the list starting at 0 for a list of simple values, and the basic storage of the underlying values. The storage cannot be either **StorageType.UNKNOWN** or **StorageType.LIST**.

```
d.createModelOutputColumn(prefix, basename, storageType, measureType, modelOutputMetadata) : Column
```

prefix (string) : the column prefix (may not contain "-" characters)

basename (string) : the column base name

storageType (StorageType) : the storage type

measureType (MeasureType) : the measure type

modelOutputMetadata (ModelOutputMetadata) : the model output metadata

Returns a new **Column** with the specified type metadata and associated model output metadata.

Exceptions:

DataModelError : if the prefix is invalid

```
d.createModelOutputColumn(prefix, basename, sourceColumn, modelOutputMetadata) : Column
```

prefix (string) : the column prefix (may not contain "-" characters)

basename (string) : the column base name

sourceColumn (Column) : the source column specifying the column type

modelOutputMetadata (ModelOutputMetadata) : the model output metadata

Returns a new **Column** with the same type metadata as the source column along with the associated model output metadata.

Exceptions:

DataModelError : if the source column was not created by the **DataModelFactory** or the prefix is invalid.

```
d.createModelOutputMetadata(modelFieldRole, targetColumn, value, group, tag) : ModelOutputMetadata
```

modelFieldRole (ModelFieldRole) : the mode column role

targetColumn (string) : the target column (may be **None**)

value (object) : the specific value associated with this column (may be **None**). If supplied, this must be an instance of **string**, **int**, or **float**.

group (List) : the column group definition (may be **None**). If supplied this must be a list on **int** values which are ≥ 1 .

tag (**string**) : the tag (may be **None**)

Returns a new model output metadata object with the specified attributes.

d.createRangeColumn(name, label, storageType, lowerBound, upperBound) : Column

name (**string**) : the column name

label (**string**) : the column label (may be **None**)

storageType (**StorageType**) : the storage definition

lowerBound (**object**) : the lower range bound

upperBound (**object**) : the upper range bound

Returns a **Column** with the supplied attributes.

d.createSetColumn(name, label, storageType, values) : Column

name (**string**) : the column name

label (**string**) : the column label (may be **None**)

storageType (**StorageType**) : the storage definition

values (**object[]**) : the valid values for the column

Returns a **Column** with the supplied attributes.

d.dataModelToXML(dataModel) : string

dataModel (**DataModel**) : the data model

Returns an XML format string containing the supplied data model in a serializable format. The data model can be recreated by calling **xmlToDataModel**.

Exceptions:

com.spss.psapi.data.DataModelException : if the data model XML cannot be created

d.extendDataModel(column, initialDataModel) : DataModel

column (**Column**) : the column

initialDataModel (**DataModel**) :

Returns a **DataModel** consisting of the column appended to the supplied data model. The supplied array must not contain **None** values.

Exceptions:

com.spss.psapi.data.DataModelException : if the column was not created by the **DataModelFactory**.

d.extendDataModel(addition, initialDataModel) : DataModel

addition (**DataModel**) : the data model to be appended

initialDataModel (**DataModel**) :

Returns a **DataModel** consisting of the initial data model with the additions appended.

Exceptions:

com.spss.psapi.data.DataModelException : if the data models were not created by the **DataModelFactory**.

d.extendDataModel(columns, initialDataModel) : DataModel

columns (**Column[]**) : the columns

initialDataModel (**DataModel**) :

Returns a **DataModel** consisting of the columns appended to the supplied data model. The supplied array must not contain **None** values.

Exceptions:

com.spss.psapi.data.DataModelException : if the columns array contains **None** values.

d.extractDataModel(names, initialDataModel) : DataModel

names (**string[]**) : the column names

`initialDataModel (DataModel) : the source data model`

Returns a `DataModel` consisting of the named columns extracted from the source data model. If the source data model does not include a specified column name, that column will be ignored.

Exceptions:

`com.spss.psapi.data.DataModelException` : if the array contains `None` values.

`d.extractDataModel(storageTypes, initialDataModel) : DataModel`

`storageTypes (StorageType[])` : the storage types

`initialDataModel (DataModel) : the source data model`

Returns a `DataModel` consisting of columns in the source data model that have the supplied storage type(s).

Exceptions:

`com.spss.psapi.data.DataModelException` : if the array contains `None` values.

`d.extractDataModel(measureTypes, initialDataModel) : DataModel`

`measureTypes (MeasureType[])` : the measure types

`initialDataModel (DataModel) : the source data model`

Returns a `DataModel` consisting of columns in the source data model that have the supplied measure type(s).

Exceptions:

`com.spss.psapi.data.DataModelException` : if the array contains `None` values.

`d.extractDataModel(modelingRoles, initialDataModel) : DataModel`

`modelingRoles (ModelingRole[])` : the modeling roles types

`initialDataModel (DataModel) : the source data model`

Returns a `DataModel` consisting of columns in the source data model that have the supplied modeling role(s).

Exceptions:

`com.spss.psapi.data.DataModelException` : if the array contains `None` values.

`d.modifyColumns(modifiedColumns) : DataModel`

`modifiedColumns (Collection)` : the columns to be modified

Creates and returns a new data model based on this data model with the specified columns modified. The supplied columns will be used to replace any existing column with the same name meaning that this method cannot be used to change the name of columns in the data model. If a supplied column name does not match an existing column, it will be ignored.

`d.removeColumns(columnNames) : DataModel`

`columnNames (Collection)` :

Creates and returns a new data model based on this data model with the specified columns modified. The supplied columns will be used to replace any existing column with the same name meaning that this method cannot be used to change the name of columns in the data model. If a supplied column does not match an existing column, it will be added to the end of the data model.

Exceptions:

`DataModelError` : if the modified columns are invalid

`d.removeFromDataModel(names, initialDataModel) : DataModel`

`names (String[])` : the column names

`initialDataModel (DataModel) : the source data model`

Returns a `DataModel` consisting of the source data model with the named columns removed.

Exceptions:

`com.spss.psapi.data.DataModelError` : if the array contains `None` values.

`d.removeFromDataModel(storageTypes, initialDataModel) : DataModel`

`storageTypes (StorageType[])` : the storage types

`initialDataModel (DataModel)` : the source data model

Returns a `DataModel` consisting of the source data model with columns of the supplied storage type(s) removed.

Exceptions:

`com.spss.psapi.data.DataModelError` : if the array contains `None` values.

`d.removeFromDataModel(measureTypes, initialDataModel) : DataModel`

`measureTypes (MeasureType[])` : the measure types

`initialDataModel (DataModel)` : the source data model

Returns a `DataModel` consisting of the source data model with columns of the supplied measure type(s) removed.

Exceptions:

`com.spss.psapi.data.DataModelError` : if the array contains `None` values.

`d.removeFromDataModel(modelingRoles, initialDataModel) : DataModel`

`modelingRoles (ModelingRole[])` : the modeling roles types

`initialDataModel (DataModel)` : the source data model

Returns a `DataModel` consisting of the source data model with columns of the supplied modeling roles(s) removed.

Exceptions:

`com.spss.psapi.data.DataModelError` : if the array contains `None` values.

`d.renameColumns(modifiedNames) : DataModel`

`modifiedNames (Map)` : a map containing the original column names as the key and the new name as the associated value

Creates and returns a new data model based on this data model with the specified columns renamed. If the map contains keys that do not correspond to columns in the data model, the key will be ignored.

Exceptions:

`DataModelError` : if the new column names are invalid, for example cause the data model to have duplicate names.

`d.toDataModel(columns) : DataModel`

`columns (Column[])` : Returns a `DataModel` containing the supplied columns. The columns are inserted in array order.

Exceptions:

`DataModelError` : if the source columns are not the system defined implementation

`d.xmlToDataModel(xml) : DataModel`

`xml (string)` : the data model XML

Returns a `DataModel` defined by the supplied XML string. It is assumed the string was generated by calling `dataModelToXML`.

Exceptions:

`DataModelError` : if the data model cannot be recreated

Related information

- [Data and Metadata](#)
-

ExtendedMeasure Objects

This is an empty marker interface which identifies classes that provide extended measure metadata.

Related information

- [Data and Metadata](#)

ExtendedStorage Objects

This is an empty marker interface which identifies classes that provide extended storage metadata.

Related information

- [Data and Metadata](#)

GeoType Objects

Subclass of **ExtendedMeasure**.

Defines the different categories of geo objects.

g.getCoordinates() : int

Returns a positive integer, usually 2 or 3, which is the number of coordinates that define a point.

g.getCsname() : string

Returns a string containing the name of the coordinate system if the “well-known ID” is ≥ 0 or **None**.

g.getGeometryType() : GeometryType

Returns the type of object represented by this definition.

g.getWkid() : int

Returns an integer representing the “well-known ID” of the coordinate system or -1. If the attribute is -1, then the coordinate system is assumed to be a non-standard one.

g.getWkt() : string

Returns a string containing the “well-known text” for the coordinate system corresponding to the “well-known ID”.

g.isProjected() : boolean

For valid “well-known ID” values, this indicates whether the coordinate system is projected or geographic.

Related information

- [Data and Metadata](#)

GeometryType Objects

Defines the different categories of geo objects.

Constants:

LineString (GeometryType) : Defines that the values describe lines.

MultiLineString (GeometryType) : Defines that the values describe multiple lines.

MultiPoint (GeometryType) : Defines that the values describe multiple points.

MultiPolygon (GeometryType) : Defines that the values describe multiple polygons.

Point (GeometryType) : Defines that the values describe points.

Polygon (GeometryType) : Defines that the values describe polygons.

g.getDepth() : int

Returns the list depth that this geometry type uses.

```
valueOf(name) : GeometryType
name(string) :
values() : GeometryType[]
```

Related information

- [Data and Metadata](#)

GlobalValues Objects

This defines the set of global values which are usually computed by a stream via a "setglobals" node.

```
g.fieldNameIterator() : Iterator
```

Returns an iterator for each field name with at least one global value.

```
g.getValue(type, fieldName) : Object
```

type (GlobalValues.Type) : the type of value

fieldName (string) : the field name

Returns the global value for the specified type and field name or **None** if no value can be located. The returned value is generally expected to be a number although future functionality may return different value types.

```
g.getValues(fieldName) : Map
fieldName (string) :
```

Returns a map containing the known entries for the specified field name or **None** of no entries for the field exist.

Related information

- [Data and Metadata](#)

GlobalValues.Type Objects

This defines the set of global value types than can be accessed from the global values table.

Constants:

- MAX (GlobalValues.Type) :
- MEAN (GlobalValues.Type) :
- MIN (GlobalValues.Type) :
- STDDEV (GlobalValues.Type) :
- SUM (GlobalValues.Type) :

```
g.getFunctionName() : string
valueOf(name) : GlobalValues.Type
name(string) :
values() : GlobalValues.Type[]
```

Related information

- [Data and Metadata](#)

InvalidColumnExceptionValues Objects

Subclass of **DataModelError**.

An **InvalidColumnValueException** is thrown when the value supplied for a column is not consistent with its storage type.

No message string is set for this exception.

`i.getColumnName() : string`
Returns the name of the unknown column.

`i.getValue() : Object`
Returns the invalid value.

Related information

- [Data and Metadata](#)
-

ListStorage Objects

Subclass of `ExtendedStorage`.

Defines the storage metadata associated with lists

`l.getDepth() : int`
Returns the depth of the list. A list consisting of simple values has a depth of 0.

`l.getValueStorage() : StorageType`
Returns the simple storage of the values held in the list. The return value will not be `Storage.UNKNOWN` or `Storage.LIST`.

Related information

- [Data and Metadata](#)
-

MeasureType Objects

Subclass of `Enum`.

This class enumerates the valid measures for columns in a `DataModel`.

Constants:

- `AUTOMATIC (MeasureType)` : Indicates that the column value will be auto-selected as RANGE or DISCRETE according to its storage.
- `COLLECTION (MeasureType)` : Indicates that the column value should be interpreted as a collection of values.
- `DISCRETE (MeasureType)` : Indicates that the column value is like SET but can be treated as FLAG if there are only two values.
- `FLAG (MeasureType)` : Indicates that the column value is one of two values.
- `GEOSPATIAL (MeasureType)` : Indicates that the column value should be interpreted as a geospatial value.
- `ORDERED_SET (MeasureType)` : Indicates that the column value is like SET but with an implied order on the values.
- `RANGE (MeasureType)` : Indicates that the column value lies between two points on a scale.
- `SET (MeasureType)` : Indicates that the column value is one of a (small) set of discrete values.
- `TYPELESS (MeasureType)` : Indicates that the column can have any value compatible with its storage.

`getEnum(name) : MeasureType`
`name (string) : the enumeration name`
Returns the enumeration with the supplied name or `None` if no enumeration exists for the supplied name.

`getValues() : MeasureType[]`
Returns an array containing all the valid values for this enumeration class.

Related information

- [Data and Metadata](#)
-

MissingValueDefinition Objects

This defines the attributes associated with missing values.

`m.getValueCount() : int`

Returns the number of values specified as missing in this definition, excluding **None** and whitespace values. The result is the length of the list returned by `getValues`.

`m.getValues() : List`

Returns the list of values specified as missing in this definition. The list cannot be modified; the mutator methods on the list throw `UnsupportedOperationException`.

`m.isEnabled() : boolean`

Returns **True** if this definition is enabled. A definition which is not enabled recognizes no values as missing.

`m.isNullIncluded() : boolean`

Returns **True** if the **None** value is recognized as missing by this definition.

`m.isWhitespaceIncluded() : boolean`

Returns **True** if whitespace values are recognized as missing by this definition.

Related information

- [Data and Metadata](#)
-

ModelOutputMetadata Objects

This defines the metadata associated with a Column generated by a model.

Constants:

- **ADJUSTED** (`string`) : Used by binary classifiers as the tag value indicating that the column indicates adjusted propensity.
- **RAW** (`string`) : Used by binary classifiers as the tag value indicating that the column indicates raw propensity.
- **X** (`string`) : Used by Kohonen models as the tag value indicating the X component of the cluster id.
- **Y** (`string`) : Used by Kohonen models as the tag value indicating the Y component of the cluster id.

`m.getRole() : ModelFieldRole`

Returns the model output column role for the column if it is a model output column or **None**.

`m.getTag() : string`

Returns the tag associated with this model output column or **None** if the column is not a model output column or the tag has not been defined.

`m.getTargetColumn() : string`

Returns the target column name associated with this model output column or **None** if the column is not a model output column or the target column has not been defined.

`m.getValue() : Object`

Returns the value associated with this model output column or **None** if the column is not a model output column or no value has been defined. The value is typically used in combination with the target column e.g. when a model can generate probabilities for each value specified by the target field.

Related information

- [Data and Metadata](#)
-

ModelingRole Objects

Subclass of `Enum`.

This class enumerates the valid modeling roles for columns in a `DataModel`.

Constants:

- **BOTH** (`ModelingRole`) : Indicates that this column can be either an antecedent or a consequent.
- **FREQ_WEIGHT** (`ModelingRole`) : Indicates that this column is used to be as frequency weight but won't be showed for user.
- **IN** (`ModelingRole`) : Indicates that this column is a predictor or an antecedent.
- **NONE** (`ModelingRole`) : Indicates that this column is not used directly during modeling.
- **OUT** (`ModelingRole`) : Indicates that this column is predicted or a consequent.
- **PARTITION** (`ModelingRole`) : Indicates that this column is used to identify the data partition.
- **RECORD_ID** (`ModelingRole`) : Indicates that this column is used to identify the record id.
- **SPLIT** (`ModelingRole`) : Indicates that this column is used to split the data.

`getEnum(name) : ModelingRole`

`name (string)` : the enumeration name

Returns the enumeration with the supplied name or `None` if no enumeration exists for the supplied name.

`getValues() : ModelingRole[]`

Returns an array containing all the valid values for this enumeration class.

Related information

- [Data and Metadata](#)

RowSet Objects

This defines the properties of a tabular dataset.

`r.getColumnClass(columnIndex) : Class`

`columnIndex (int)` : the column index

Returns the class of objects in a specified column of this dataset. The column index must be in the range:

`0 <= index < getColumnCount()`

Exceptions:

`IndexOutOfBoundsException` : unless the column index is in range

`r.getColumnCount() : int`

Returns the number of columns in this dataset.

`r.getColumnName(columnIndex) : string`

`columnIndex (int)` : the column index

Returns the name of a specified column in this dataset. Returns the empty string if the column does not have a name. The column index must be in the range:

`0 <= index < getColumnCount()`

Exceptions:

`IndexOutOfBoundsException` : unless the column index is in range

`r.getRowCount() : int`

Returns the number of rows in this dataset.

`r.getValueAt(rowIndex, columnIndex) : Object`

`rowIndex (int)` : the row index

`columnIndex (int)` : the column index

Returns the value from a specified row and column in this dataset. The row and column indexes must be in the range:

`0 <= rowIndex < getRowCount() && 0 <= columnIndex < getColumnCount()`

The returned value will either be `None` or an instance of the class returned by `getColumnClass` for the same column index.

Exceptions:

`IndexOutOfBoundsException` : unless the row and column indexes are in range

Related information

- [Data and Metadata](#)
-

StorageType Objects

Subclass of `Enum`.

This class enumerates the valid storage types for columns in a `DataModel` or parameters.

Constants:

- `DATE (StorageType)` : Indicates that the storage type is a date type.
- `INTEGER (StorageType)` : Indicates that the storage type is an integer.
- `LIST (StorageType)` : Indicates that the storage type is a list of primitive values.
- `REAL (StorageType)` : Indicates that the storage type is a floating point number.
- `STRING (StorageType)` : Indicates that the storage type is a string.
- `TIME (StorageType)` : Indicates that the storage type is a time type.
- `TIMESTAMP (StorageType)` : Indicates that the storage type is a combined time and date type.
- `UNKNOWN (StorageType)` : Indicates that the storage type is unknown.

`getEnum(name) : StorageType`

`name (string)` : the enumeration name

Returns the enumeration with the supplied name or `None` if no enumeration exists for the supplied name.

`getValues() : StorageType[]`

Returns an array containing all the valid values for this enumeration class.

Related information

- [Data and Metadata](#)
-

UnknownColumnException Objects

Subclass of `DataModelError`.

An `UnknownColumnException` is thrown when a column name is used that does not exist in the data model.

No message string is set for this exception.

`u.getColumnName() : string`

Returns the name of the unknown column.

Related information

- [Data and Metadata](#)
-

Expressions

This provides support for parsing CLEM expressions.

- [Expression Objects](#)
- [Parser Objects](#)
- [ParserException Objects](#)

Related information

- [Expression Objects](#)
 - [Parser Objects](#)
 - [ParserException Objects](#)
-

Expression Objects

This interface encapsulates details of a CLEM expression.

```
e.getReferencedFields() : set
```

Returns the **Set** of field names referenced by the expression.

```
e.getStorageType() : StorageType
```

Returns the **StorageType** of the expression.

Related information

- [Expressions](#)
-

Parser Objects

This interface encapsulates the functionality for parsing CLEM expressions.

```
p.parseExpression(expressionString, dataModel) : Expression
```

expressionString (**string**) :- the **string** to parse

dataModel (**DataModel**) :- the data model

Attempts to parse the **string** argument to an **Expression**. Parsing checks for syntax, lexical and semantic errors. The **dataModel** argument is used to check for fields referenced in the expression. Parsing using this method will fail if the **string** contains reference to **Node** or **Stream** related functions and values such as GlobalValues, Parameters and database functions.

Exceptions

ParserException: - if the **string** does not contain a parsable Expression.

Related information

- [Expressions](#)
-

ParserException Objects

Subclass of **ModelerException**.

A **ParserException** is thrown to indicate that an attempt to parse a **string** to an **Expression** failed. Parsing can fail due to syntax, lexical or semantic errors.

```
p.getEndPosition() : int
```

Returns the end position in the parsed **string** of the text relating to the error. Returns -1 if start position is not valid.

```
p.getStartPosition() : int
```

Returns the start position in the parsed **string** of the text relating to the error. Returns -1 if start position is not valid.

Related information

- [Expressions](#)
-

Model information

This provides support for accessing information about data mining models.

- [CompositeModelDetail Objects](#)
- [ModelDetail Objects](#)
- [ModelType Objects](#)
- [PMMLModelType Objects](#)

Related information

- [CompositeModelDetail Objects](#)
- [ModelDetail Objects](#)
- [ModelType Objects](#)
- [PMMLModelType Objects](#)

CompositeModelDetail Objects

Subclass of [ModelDetail](#).

This interface encapsulates the representation detail of a composite model.

`c.canUseMultipleModels() : boolean`

Returns `True` if multiple models can be used by the composite model.

`c.getIndividualModelResults() : Iterator`

Returns the individual model results of this composite model.

Related information

- [Model information](#)

ModelDetail Objects

This interface encapsulates the representation detail of a data mining model.

`m.getAlgorithmName() : string`

Returns the name of the model builder algorithm.

`m.getApplicationName() : string`

Returns the name of the model builder application.

`m.getApplicationVersion() : string`

Returns the version of the model builder application.

`m.getBuildDate() : Date`

Returns the date this model was built.

`m.getCopyright() : string`

Returns the model copyright.

`m.getInputDataModel() : DataModel`

Returns the input data model required by the model. Note that the `ModelingRole` for each field in the input data model is `ModelingRole.IN`, even for fields that had been specified as `ModelingRole.BOTH` for association or sequence models.

`m.getModelID() : string`

Returns the model ID. For models within a composite model, it is assumed that each model has a unique ID.

```
m.getModelType() : ModelType
```

Returns the type of the model.

```
m.getOutputDataModel() : DataModel
```

Returns the output data model produced by the model. Note that the **ModelingRole** for each field in the output data model is **ModelingRole.OUT**, even for fields that had been specified as **ModelingRole.BOTH** for association or sequence models.

It is also important to note that this is not same as the output column set produced by the **ModelApplier** that applies the model to data. The transformer output column set will usually include additional fields from the input and may include property settings that modify the number or type of outputs produced (for example, a transformer that applies a clustering model may produce the cluster ID as an integer or a string depending on the property settings).

```
m.getPMMLModelType() : PMMLModelType
```

For a model which is represented internally using PMML (one for which **isPMMLModel()** returns **True**) this returns the PMML model type from the underlying PMML that was generated by the modeling algorithm. For a model which is not represented using PMML this returns **None**.

```
m.getPMMLText() : string
```

For a model which is represented internally using PMML (one for which **isPMMLModel()** returns **True**) this returns a string representing the PMML generated from the model building algorithm. For non-PMML models this returns **None**. When this returns **None** you may be able to obtain an alternative representation of the model as PMML using the export function: **TaskRunner.exportModelToFile()**.

```
m.getSplitColumnNames() : List
```

Returns the list of split column names if this is a split model or **None** for non-split models.

```
m.getSplitModelCount() : int
```

Returns the number of split models if this is a split model or **0** otherwise.

```
m.getSplitModelKey(index) : List
```

index (int) : the model index

Returns the values of the split fields at the specified index for split models or **None** for non-split models. When present, the values are in the order specified by the split output columns.

```
m.getSplitModelPMMLText(index) : string
```

index (int) : the model index

Returns the PMML representation of the split model at the specified index for split models or **None** for non-split models or if the split model is not represented as PMML.

```
m.getUserName() : string
```

Returns the name of the account used to build the model.

```
m.isOutputColumnAuxiliary(name) : boolean
```

name (string) : the column name

Returns **True** if the supplied column name is an auxiliary column in the output data model or **False**. An auxiliary output column is a column that provides additional information about the output of the model, for example, the confidence of the prediction or distance from the cluster center.

```
m.isPMMLModel() : boolean
```

Returns **True** if this model is represented using PMML, or **False** otherwise.

```
m.isSplitModel() : boolean
```

Returns **True** if this model is a split model, **False** otherwise.

Related information

- [Model information](#)
-

ModelType Objects

Subclass of **Enum**.

This class enumerates the valid model types. These are largely based on the JDM 1.1 and draft JDM 2.0 data mining functions definition.

Constants:

- **ANOMALY_DETECTION** (`ModelType`) : Indicates an anomaly detection model.
- **APPROXIMATION** (`ModelType`) : Indicates that the model predicts a continuous value.
- **ASSOCIATION** (`ModelType`) : Indicates that the model identifies frequently occurring sets of items.
- **ATTRIBUTE_IMPORTANCE** (`ModelType`) : Indicates an attribute importance model.
- **CATEGORIZE** (`ModelType`) : Indicates a text categorize model.
- **CLASSIFICATION** (`ModelType`) : Indicates that the model predicts a discrete value.
- **CLUSTERING** (`ModelType`) : Indicates that the model groups similar rows of data together.
- **CONCEPT_EXTRACTION** (`ModelType`) : Indicates that the model identifies concepts from textual data.
- **REDUCTION** (`ModelType`) : Indicates that the model reduces the complexity of data.
- **SEQUENCE** (`ModelType`) : Indicates that the model identifies frequently occurring sequences of events.
- **SUPERVISED_MULTITARGET** (`ModelType`) : Indicates a supervised multi-target model.
- **SURVIVAL_ANALYSIS** (`ModelType`) : Indicates a survival analysis model.
- **TIME_SERIES** (`ModelType`) : Indicates a time series model.
- **UNKNOWN** (`ModelType`) : Indicates that the model type cannot be determined. This value should never be returned for any built-in model.

`getEnum(name) : ModelType`

`name (string)` : the enumeration name

Returns the enumeration with the supplied name or `None` if no enumeration exists for the supplied name.

`getValues() : ModelType[]`

Returns an array containing all the valid values for this enumeration class.

`m.isUnknown() : boolean`

Returns `True` if this value is `ModelType.UNKNOWN`.

Related information

- [Model information](#)
-

PMMMLModelType Objects

Subclass of `Enum`.

Enumerates the model types specified by PMML 4.0.

Constants:

- **ASSOCIATION** (`PMMMLModelType`) : PMML AssociationModel
- **CLUSTERING** (`PMMMLModelType`) : PMML ClusteringModel
- **GENERAL_REGRESSION** (`PMMMLModelType`) : PMML GeneralRegressionModel
- **MINING** (`PMMMLModelType`) : PMML MiningModel
- **NAIVE_BAYES** (`PMMMLModelType`) : PMML NaiveBayesModel
- **NEURAL_NETWORK** (`PMMMLModelType`) : PMML NeuralNetwork
- **REGRESSION** (`PMMMLModelType`) : PMML RegressionModel
- **RULE_SET** (`PMMMLModelType`) : PMML RuleSetModel
- **SEQUENCE** (`PMMMLModelType`) : PMML SequenceModel
- **SUPPORT_VECTOR_MACHINE** (`PMMMLModelType`) : PMML SupportVectorMachineModel
- **TEXT** (`PMMMLModelType`) : PMML TextModel
- **TIME_SERIES** (`PMMMLModelType`) : PMML TimeSeriesModel
- **TREE** (`PMMMLModelType`) : PMML TreeModel

`getEnum(name) : PMMMLModelType`

`name (string)` : the PMML model type name

Returns the `PMMMLModelType` corresponding to the specified PMML model type name, or `None` if there is no such model type. The name must be one specified by PMML 4.0.

`getValues() : PMMMLModelType[]`

Returns an array containing all the valid values for this enumeration class.

```
p.isUnknown() : boolean
```

Returns **False** always. All the values in this class represent known model types; there is no "unknown".

Related information

- [Model information](#)

Server resources

This provides access to server-side resources such as the server file system and database connections.

- [ServerDatabaseConnection Objects](#)
- [ServerFile Objects](#)
- [ServerFileSystem Objects](#)
- [ServerResourceException Objects](#)

Related information

- [ServerDatabaseConnection Objects](#)
- [ServerFile Objects](#)
- [ServerFileSystem Objects](#)
- [ServerResourceException Objects](#)

ServerDatabaseConnection Objects

This encapsulates the functionality of an object that represents a connection to a database.

```
s.dropTable(tableName)
tableName (string) :
s.getDBQueryColumns(queryText) : RowSet
queryText (string) : the text of the query
```

Returns a list of the columns returned by a specified database query. The list is returned in the form of a dataset where the first column contains the column name as a string.

Exceptions:

```
ServerResourceException : if the server resource cannot be accessed
s.getDBTableColumns(catalogName, schemaName, tableName) : RowSet
catalogName (string) : the catalog name
schemaName (string) : the schema name
tableName (string) : the base table name
```

Returns a list of the columns in a specified database table. The list is returned in the form of a dataset where the first column contains the column name as a string and the second column contains **StorageType** constants that provide the closest match for the storage types for the columns in the database table.

Exceptions:

```
ServerResourceException : if the server resource cannot be accessed
s.getDBTables(catalogName, schemaName, tableName, includeUserTables, includeSystemTables, includeViews,
includeSynonyms) : RowSet
catalogName (string) : pattern for the catalog name
schemaName (string) : pattern for the schema name
tableName (string) : pattern for the table name
includeUserTables (boolean) : indicates whether user tables should be included
```

```
includeSystemTables (boolean) : indicates whether system tables should be included
```

```
includeViews (boolean) : indicates whether views should be included
```

```
includeSynonyms (boolean) : indicates whether synonyms should be included
```

Returns a summary of the tables available on this database connection which match the specified arguments. The summary is returned in the form of a dataset with at least two columns, of which the first two columns are the schema name and table name, both as strings.

Exceptions:

```
ServerResourceException : if the server resource cannot be accessed
```

```
s.isDBTableVisible(catalogName, schemaName, tableName) : boolean
```

```
catalogName (string) : the catalog name
```

```
schemaName (string) : the schema name
```

```
tableName (string) : the table name
```

Returns **True** if the specified table is visible to this connection.

Exceptions:

```
ServerResourceException : if the server resource cannot be accessed
```

Related information

- [Server resources](#)

ServerFile Objects

This encapsulates the representation of a file on the server file system. A server file always uses / to separate directory paths since this is valid for both UNIX and Windows file systems.

```
s.getName() : string
```

Returns the name of the file.

```
s.getParent() : string
```

Returns the pathname of the parent directory.

```
s.getParentServerFile() : ServerFile
```

Returns a file representing the parent directory.

```
s.getPath() : string
```

Returns the pathname of the file.

```
s.isAbsolute() : boolean
```

Returns whether the pathname is absolute.

```
s.isDirectory() : boolean
```

Returns whether this file is a directory.

Related information

- [Server resources](#)

ServerFileSystem Objects

This encapsulates the functionality for representing the file system on the data mining server's host.

```
s.createTemporaryFile(name) : ServerFile
```

`name (string)` : the suggested name of the new file

Creates a new file with a unique name in the server's temporary file space. The file is empty and can be written by the creator. The specified name will be used as the starting point for generating a unique name for the file, and if the name has an extension it will be left unchanged. The file must be deleted when it is no longer needed.

Exceptions:

`ServerResourceException` : if the file cannot be created for any reason

`s.deleteFile(file)`

`file (ServerFile)` : the file to delete

Deletes the specified file from this file system. The caller must have permission to delete the file.

Calling this method has the same effect as creating and running a delete file task.

Exceptions:

`ServerResourceException` : if the file cannot be deleted

`s.directoryOrFileExists(path, isDirectory) : boolean`

`path (ServerFile)` : Pathname to check

`isDirectory (boolean)` : Set to True if the path to be tested should be a directory

Returns whether a directory or file exists.

Exceptions:

`ServerResourceException` : if the server file system cannot be accessed

`s.exists(file) : boolean`

`file (ServerFile)` :

Returns whether a file exists.

Exceptions:

`ServerResourceException` : if the server file system cannot be accessed

`s.getAbsolutePath(file) : string`

`file (ServerFile)` : a `ServerFile` object

Returns the absolute pathname or `None` if the absolute path cannot be determined.

Exceptions:

`ServerResourceException` : if the server file system cannot be accessed

`s.getAbsoluteServerFile(file) : ServerFile`

`file (ServerFile)` : a `ServerFile` object

Returns a file representing the absolute pathname or `None` if the absolute path cannot be determined.

Exceptions:

`ServerResourceException` : if the server file system cannot be accessed

`s.getChild(parent, filename) : ServerFile`

`parent (ServerFile)` : a `ServerFile` object representing a directory or special folder

`filename (string)` : a name of a file or folder which exists in `parent`

Returns the named child in the supplied parent.

Exceptions:

`ServerResourceException` : if the server file system cannot be accessed

`s.getDefaultDirectory() : ServerFile`

Returns the server's default directory.

Exceptions:

ServerResourceException : if the server file system cannot be accessed

s.GetFiles(directory) : ServerFile[]

directory (ServerFile) : the directory

Returns the list of files in the specified directory of the file system.

Exceptions:

ServerResourceException : if the server file system cannot be accessed

s.getParentDirectory(file) : ServerFile

file (ServerFile) : a **ServerFile** object

Returns the parent directory of a file in the file system.

s.getPathSeparator() : string

Returns a string version of the path separator character.

s.getPathSeparatorChar() : char

Returns the path separator character for this file system. On Windows, this is ; and on UNIX, this is :.

s.getRoots() : ServerFile[]

Returns the roots of the file system.

Exceptions:

ServerResourceException : if the server file system cannot be accessed

s.getSeparator() : string

Returns a string version of the separator character.

s.getSeparatorChar() : char

Returns the separator character for this file system. On Windows, this is \ and on UNIX, this is /.

s.getServerFile(filename) : ServerFile

filename (string) : a name of a file or folder

Returns a server file for the corresponding file name.

s.getSize(file) : long

file (ServerFile) : the file

Returns the size, in bytes, of the specified file in this file system. Returns 0 if the file does not exist or is not accessible, or -1 if the file is accessible but the server does not support the file size operation. The result is undefined when the file denotes a directory.

Exceptions:

ServerResourceException : if the file system cannot be accessed

Related information

- [Server resources](#)
-

ServerResourceException Objects

Subclass of **ModelerException**.

An exception thrown while accessing a server resource.

A message string is always created for this exception.

Related information

-
- [Server resources](#)

Server resources

This provides support for the construction and use of data mining sessions.

- [LocaleInfo Objects](#)
- [Repository Objects](#)
- [Session Objects](#)
- [SessionException Objects](#)
- [SystemSession Objects](#)
- [UIResources Objects](#)

Related information

- [LocaleInfo Objects](#)
- [Repository Objects](#)
- [Session Objects](#)
- [SessionException Objects](#)
- [SystemSession Objects](#)
- [UIResources Objects](#)

LocaleInfo Objects

This interface defines locale-sensitive information associated with a **Session**.

```
l.getLocale() : Locale
```

Returns the locale associated with this locale information object.

```
l.getLocalizedProcessorDescription(nodeType) : string
```

```
nodeType (string) : the node type
```

Returns a locale-sensitive string describing the supplied node type name or **None** if a description cannot be found.

```
l.getLocalizedProcessorName(nodeType) : string
```

```
nodeType (string) : the node type name
```

Returns a locale-sensitive string representing the supplied node type name or **None** if a name cannot be found.

Related information

- [Server resources](#)

Repository Objects

This defines the basic functionality for a content repository. Unlike the file persistence tasks defined by **TaskFactory**, these are executed synchronously through the repository object rather than indirectly via the Session.

```
r.createFolder(parentFolder, newFolder) : string
```

```
parentFolder (string) : the path to the parent folder
```

```
newFolder (string) : the new folder name
```

Creates a new folder with the specified name.

Exceptions:

```
SessionException : if the folder cannot be created
```

```
r.createRetrieveURI(path, version, label) : URI
```

path (string) : the full path to the object to be retrieved

version (string) : the version marker or **None**

label (string) : the label or **None**

A utility function to create a repository URI that is valid for retrieving an object from the specified location. Either a version or label may be specified if a specific version is required. If both are **None** then the LATEST version will be returned.

Exceptions:

URISyntaxException : if the method cannot construct a valid URI

```
r.createStoreURI(path, label) : URI
```

path (string) : the full path to the store location

label (string) : the label to be applied to the object or **None**

A utility function to create a repository URI that is valid for storing an object at the specified location and with an optional label to be applied to the object when it is stored.

Exceptions:

URISyntaxException : if the method cannot construct a valid URI

```
r.deleteFolder(folder)
```

folder (string) : the folder to be deleted

Deletes the specified folder and any content within it.

Exceptions:

SessionException : if the folder cannot be deleted

```
rgetRepositoryHandle() : Object
```

Returns the underlying repository handle. The handle will be an instance of the Java class
`com.spss.repository.client.application.Repository`.

```
r.renameFolder(folder, newName)
```

folder (string) : the path of the folder to be renamed

newName (string) : the new folder name

Renames the specified folder.

Exceptions:

SessionException : if the folder cannot be renamed

```
r.retrieveDocument(path, version, label, autoManage) : DocumentOutput
```

path (string) : the full path to the object

version (string) : the version marker or **None**

label (string) : the label or **None**

autoManage (boolean) : whether the document should be added to the output manager

Retrieves an output document (in COU format) from the specified path. Either a version or label may be specified if a specific version is required. If both are **None** then the LATEST version is returned. Code that needs to open documents privately without having them made visible to the user should set the **autoManage** flag to **False**.

Exceptions:

URISyntaxException : if the method cannot construct a valid URI

SessionException : if the document output cannot be retrieved for some reason

```
r.retrieveModel(path, version, label, autoManage) : ModelOutput
```

path (string) : the full path to the object

`version (string) : the version marker or None`

`label (string) : the label or None`

`autoManage (boolean) : whether the model should be added to the model manager`

Retrieves a model output from the specified path. Either a version or label may be specified if a specific version is required. If both are **None** then the LATEST version is returned. Code that needs to open models privately without having them made visible to the user should set the `autoManage` flag to **False**.

Exceptions:

`URISyntaxException : if the method cannot construct a valid URI`

`SessionException : if the model output cannot be retrieved for some reason`

`r.retrieveProcessor(path, version, label, diagram) : Node`

`path (string) : the full path to the object`

`version (string) : the version marker or None`

`label (string) : the label or None`

`diagram (Diagram) : the diagram that the node should be added to`

Retrieves a node from the specified path and inserts it into the supplied diagram. Either a version or label may be specified if a specific version is required. If both are **None** then the LATEST version is returned.

Exceptions:

`URISyntaxException : if the method cannot construct a valid URI`

`SessionException : if the node cannot be retrieved for some reason`

`r.retrieveStream(path, version, label, autoManage) : Stream`

`path (string) : the full path to the object`

`version (string) : the version marker or None`

`label (string) : the label or None`

`autoManage (boolean) : whether the stream should be added to the stream manager`

Retrieves a stream from the specified path. Either a version or label may be specified if a specific version is required. If both are **None** then the LATEST version is returned. Code that needs to open streams privately without having them made visible to the user should set the `autoManage` flag to **False**.

Exceptions:

`URISyntaxException : if the method cannot construct a valid URI`

`SessionException : if the stream cannot be retrieved for some reason`

`r.storeDocument(documentOutput, path, label) : string`

`documentOutput (DocumentOutput) : the document output to be stored`

`path (string) : the path`

`label (string) : the label or None`

Stores a document output to the specified location. If the label is provided then it is applied to the new version.

Exceptions:

`URISyntaxException : if the method cannot construct a valid URI`

`SessionException : if the document output cannot be stored for some reason`

`r.storeModel(modelOutput, path, label) : string`

`modelOutput (ModelOutput) : the model output to be stored`

`path (string) : the path`

`label (string) : the label or None`

Stores a model output to the specified location. If the label is provided then it is applied to the new version.

Exceptions:

URISyntaxException : if the method cannot construct a valid URI

SessionException : if the model output cannot be stored for some reason

r.storeProcessor(node, path, label) : string

node (Node) : the node to be stored

path (string) : the path

label (string) : the label or **None**

Stores a node to the specified location. If the label is provided then it is applied to the new version.

Exceptions:

URISyntaxException : if the method cannot construct a valid URI

SessionException : if the node cannot be stored for some reason

r.storeStream(stream, path, label) : string

stream (Stream) : the stream to be stored

path (string) : the path

label (string) : the label or **None**

Stores a stream to the specified location. If the label is provided then it is applied to the new version.

Exceptions:

URISyntaxException : if the method cannot construct a valid URI

SessionException : if the stream cannot be stored for some reason

Related information

- [Server resources](#)
-

Session Objects

Subclass of **SystemSession**.

A session is the main interface through which users of the Modeler API access the API features. Each session has its own connection to a data mining server.

A data mining server is typically a remote server identified by a **ServerConnectionDescriptor**. From Modeler API 2.0 a server can also be a local server instance spawned as a child process of the Modeler API host process (see `connect()`). This local server mode requires a local server installation, such as a Modeler client installation, on the local machine. The installation directory is located through the system property

com.spss.psapi.session.serverInstallationDirectory

This property must point to a valid installation directory and must be set before the session factory is instantiated otherwise connections to the local server will fail.

s.close()

Closes this session. This automatically interrupts any task currently running, closes any current **Stream** and invalidates any other objects created by this session.

s.connect(serverDescriptor)

serverDescriptor (ServerConnectionDescriptor) : identifies a remote data mining server

Connects this session to the remote server identified by the specified server descriptor.

Exceptions:

ServerConnectionException : if a connection to the server cannot be established or if the Session already has a connection
s.connect()

Connects this session to a local server instance. Spawns a new local server process.

Exceptions:

ServerConnectionException : if a local server instance cannot be created or if the session already has a connection

s.connect(stream)

stream (Stream) : the stream to be connected

Connects the stream to the session's server.

Exceptions:

ServerConnectionException : if the Session is not connected or if the stream is already connected

OwnerException : if the stream was not created by this session

s.createServerDatabaseConnection(datasourceName, credentialName, catalogName) :
ServerDatabaseConnection

datasourceName (string) : the datasource name

credentialName (string) : the stored credential name

catalogName (string) : the catalog name

Creates a **ServerDatabaseConnection**. The datasource name must visible to the data mining server.

Exceptions:

ServerConnectionException : if the session is not connected to a server

ServerResourceException : if the connection task fails.

s.createServerDatabaseConnection(datasourceName, userName, password, catalogName) :
ServerDatabaseConnection

datasourceName (string) : the datasource name

userName (string) : the user name

password (string) : the password

catalogName (string) : the catalog name

Creates a **ServerDatabaseConnection**. The datasource name must visible to the data mining server.

Exceptions:

ServerConnectionException : if the session is not connected to a server

ServerResourceException : if the connection task fails.

s.getASCredentialDescriptor() : ASCredentialDescriptor

Returns the credential descriptor used to log in to the Analytic Server.

s.getAttribute(name) : Object

name (string) : an attribute name

Returns the value of the specified attribute in this session, or **None** if there is no such attribute. Attributes are created by the application, not by the Modeler API.

s.getAttributeNames() : Collection

Returns the names of the attributes defined in this session.

s.getParameters() : ParameterProvider

Returns the parameter provider for this session. The parameter provider provides access to session parameters.

s.getParser() : Parser

Returns the shared parser for this session.

```
s.getRepository() : Repository
```

Returns the repository object that provides simple mechanisms for storing and retrieving objects.

```
s.getRepositoryConnectionDescriptor() : RepositoryConnectionDescriptor
```

Returns the repository connection descriptor.

```
s.getServerConnectionDescriptor() : ServerConnectionDescriptor
```

Returns the `ServerConnectionDescriptor` used to connect this session to a remote server. Returns `None` if the session is connected to a local server or has not yet been connected.

```
s.getServerDataSourceNames() : RowSet
```

Returns a row set that lists the available system DSNs visible on the data mining server host. The list is returned in the form of a row set where the first column contains the data source name as a string and the second column contains a string description.

Exceptions:

```
ServerConnectionException : if the session is not connected to a server
```

```
ServerResourceException : if access to the server data sources was denied
```

```
s.getServerFileSystem() : ServerFileSystem
```

Returns the server file system.

Exceptions:

```
ServerConnectionException : if the session is not connected to a server
```

```
ServerResourceException : if access to the server file system was denied
```

```
s.getServerVersionInfo() : ServerVersionInfo
```

Returns information about the connected server version or `None` if the session is not connected.

```
s.getTaskFactory() : TaskFactory
```

Returns the `TaskFactory` for this session.

```
s.getTaskRunner() : TaskRunner
```

Returns the `TaskRunner` for this session.

```
s.getUIResources() : UIResources
```

Returns the interface that provides access to UI-related resources.

```
s.interrupt()
```

Interrupts any synchronous task which is currently executing. The call returns immediately while the task is interrupted. It has no effect if the session is not busy.

This method may be called from any thread.

```
s.isBusy() : boolean
```

Returns `True` if this session is currently executing a synchronous task (even if the task is in the process of being interrupted).

This method may be called from any thread.

```
s.isClosed() : boolean
```

Returns `True` if this has been closed.

```
s.isConnected() : boolean
```

Returns `True` if this session has a server connection.

```
s.isValidASCredentialDescriptor(asConnectionDescriptor) : boolean
```

```
asConnectionDescriptor (ASCredentialDescriptor) : the credential descriptor
```

Check whether the credential descriptor is a valid.

```
s.isValidRepositoryConnectionDescriptor(descriptor) : boolean
```

```
descriptor (RepositoryConnectionDescriptor) : the repository connection descriptor
```

Returns **True** if the supplied repository connection descriptor is valid i.e. can connect to a content repository.

```
s.publish(node) : ExecutionHandle  
node (DataWriter) : the DataWriter to be published
```

Executes the specified **DataWriter** in publish mode to obtain a **PublishedImage**. This overrides the execution mode set in the node and retrieves the published image from the server on completion. Execution is synchronous. The result is an **ExecutionHandle** which can be used to determine the exit status of the task and, if the task completes successfully, to obtain the result of the task which is a **PublishedImage**.

Exceptions:

OwnerException : if the node was not created by this session

ObjectLockedException : if the node or its containing stream is locked

ServerConnectionException : if the stream is not connected to a server

SessionException : if the session is already running another task, or cannot execute the task for some other reason, or if execution completes in a state other than **SUCCESS**

```
s.publish(node, inline) : ExecutionHandle
```

node (TerminalNode) : the TerminalNode to be published. Must be a **DataWriter** if not publishing inline

inline (boolean) : **True** to prepare the image for inline scoring

Executes the specified node in publish mode to obtain a **PublishedImage** ready for inline scoring.

If inline is **False**, this is equivalent to calling **publish (DataWriter)** and the node must be a data writer. Otherwise, the node may be any terminal node and the stream is modified in place to replace the input and output nodes with ones suitable for inline scoring.

Execution is synchronous. The result is an **ExecutionHandle** which can be used to determine the exit status of the publishing task and, if the task completes successfully, to obtain the result of the task which is a **PublishedImage**.

Exceptions:

OwnerException : if the node was not created by this session

ObjectLockedException : if the node or its containing stream is locked

ServerConnectionException : if the stream is not connected to a server

SessionException : if the session is already running another task, or cannot execute the task for some other reason, or if the stream cannot be prepared for inline scoring, or if execution completes in a state other than **SUCCESS**

```
s.removeAttribute(name)
```

name (string) : an attribute name

Deletes the specified attribute from this session. If there is no such attribute, the call has no effect.

```
s.run(stream, results) : ExecutionHandle
```

stream (Stream) : the Stream to be executed

results (Collection) : an empty collection that will contain any built objects once execution has completed

Executes the supplied stream synchronously and waits for it to complete. Returns an **ExecutionHandle** which can be used to access the exit status and any result from the task.

Exceptions:

OwnerException : if the stream was not created by this session

ObjectLockedException : if the stream is locked

ServerConnectionException : if the stream is not connected to a server

SessionException : if the session is already running another task, cannot execute the task or if execution completes in a state other than **SUCCESS**

```
s.run(nodes, results) : ExecutionHandle
```

nodes (Node[]) : the array of **Node** objects to be executed

results (Collection) : an empty collection that will contain any built objects once execution has completed

Executes the supplied array of nodes synchronously and waits for them to complete. There must be at least one node in the array. Returns an **ExecutionHandle** which can be used to access the exit status and any result from the task.

Exceptions:

OwnerException : if the nodes' stream was not created by this session or the nodes are not all owned by the same stream

ObjectLockedException : if the nodes' owner stream is locked

ServerConnectionException : if the nodes' stream is not connected to a server

SessionException : if the session is already running another task, cannot execute the task or if execution completes in a state other than **SUCCESS**

IllegalArgumentException : if the array is empty

s.runTask(task) : ExecutionHandle

task (Task) : the Task to be executed

Executes the supplied task synchronously and waits for it to complete. Returns an **ExecutionHandle** which can be used to access the exit status and any result from the task.

Exceptions:

OwnerException : if the task was not created by this session's **TaskFactory**

ObjectLockedException : if the task is already executing or any object referenced by the task is locked for updating

SessionException : if the session cannot execute the task or if execution completes in a state other than **SUCCESS**

s.setASCredentialDescriptor(asConnectionDescriptor)

asConnectionDescriptor (ASCredentialDescriptor) : the credential descriptor

Sets the credential descriptor used to log in Analytic Server.

s.setAttribute(name, value)

name (string) : an attribute name

value (Object) : the attribute value

Assigns a value to the specified attribute in this session. If the attribute already has a value, it is replaced. If the value is **None**, the attribute is deleted from the session.

s.setRepositoryConnectionDescriptor(descriptor)

descriptor (RepositoryConnectionDescriptor) : the new repository connection descriptor

Sets the repository connection descriptor to be used when a connection to a content repository is required. Any existing connection is closed although a new connection will not be created until it is required.

s.spawn(stream, results) : ExecutionHandle

stream (Stream) : the Stream to be executed

results (Collection) : an empty collection that will contain any built objects once execution has completed

Executes the supplied stream asynchronously. Returns an **ExecutionHandle** which can be used to monitor and control the progress of the task.

Exceptions:

OwnerException : if the stream was not created by this session

ObjectLockedException : if the stream is locked

ServerConnectionException : if the stream is not connected to a server

s.spawn(nodes, builtObjects) : ExecutionHandle

nodes (Node[]) : the array of Node objects to be executed

builtObjects (Collection) : an empty collection that will be populated by built objects created by the execution

Executes the supplied array of nodes asynchronously. Returns an **ExecutionHandle** which can be used to monitor and control the progress of the task. There must be at least one node in the array.

Exceptions:

OwnerException : if the nodes' stream was not created by this session or the nodes are not all owned by the same stream

ObjectLockedException : if the nodes' stream is locked

ServerConnectionException : if the nodes' stream is not connected to a server

IllegalArgumentException : if the array is empty

s.spawnPublish(node) : ExecutionHandle

node (DataWriter) : the **DataWriter** to be published

Executes the specified **DataWriter** asynchronously in publish mode to obtain a **PublishedImage**. This overrides the execution mode set in the node and retrieves the published image from the server on completion. Returns an **ExecutionHandle** which can be used to monitor and control the progress of the publishing task and, if it completes successfully, to obtain the task result which is a **PublishedImage**.

Exceptions:

OwnerException : if the node was not created by this session

ObjectLockedException : if the node or its containing stream is locked

ServerConnectionException : if the stream is not connected to a server

s.spawnPublish(node, inline) : ExecutionHandle

node (TerminalNode) : the **TerminalNode** to be published. Must be a **DataWriter** if not publishing inline

inline (boolean) : **True** to prepare the image for inline scoring

Executes the specified node asynchronously in *publish* mode to obtain a **PublishedImage** ready for inline scoring.

If **inline** is **False**, this is equivalent to calling **publish(DataWriter)** and the node must be a data writer. Otherwise, the node may be any terminal node and the stream is modified in place to replace the input and output nodes with ones suitable for inline scoring.

Returns an **ExecutionHandle** which can be used to monitor and control the progress of the publishing task and, if it completes successfully, to obtain the task result which is a **PublishedImage**.

Exceptions:

OwnerException : if the node was not created by this session

ObjectLockedException : if the node or its containing stream is locked

ServerConnectionException : if the stream is not connected to a server

SessionException : if the stream cannot be prepared for inline scoring

s.spawnTask(task) : ExecutionHandle

task (Task) : the **Task** to be executed

Executes the supplied task asynchronously. Returns an **ExecutionHandle** which can be used to monitor and control the progress of the task.

Exceptions:

OwnerException : if the task was not created by this session's **TaskFactory**

ObjectLockedException : if the task is already executing or any object referenced by the task is locked for updating

s.waitForAllTasksToFinish()

Waits for all tasks in progress on this session to complete.

Related information

- [Server resources](#)
-

SessionException Objects

Subclass of **ModelerException**.

An exception thrown while creating a **Session** or performing a task within a **Session**.

A message string is always created for this exception.

Related information

- [Server resources](#)
-

SystemSession Objects

Defines the functionality of a system session. Sessions define the basic context that other operations work within.

```
sgetLocale() : Locale
```

Returns the locale specified for this session.

```
s.getLocaleInfo() : LocaleInfo
```

Returns the `LocaleInfo` for this session.

```
s.isLocalSession() : boolean
```

Returns `True` if the session is connected to a local or desktop execution engine, `False` otherwise.

Related information

- [Server resources](#)
-

UIResources Objects

This interface provides access to UI-related resources.

```
u.getBaseProcessorIcon(nodeType) : ImageIcon
```

`nodeType` (`string`) : the node type

Returns the base icon used to represent this node type or `None`. The base icon does not include any background.

```
u.getProcessorIcon(nodeType) : ImageIcon
```

`nodeType` (`string`) : the node type

Returns the default icon used to represent this node type or `None`.

Related information

- [Server resources](#)
-

Tasks and execution

This provides objects that create and represent data mining tasks.

- [ExecutionFeedbackEvent Objects](#)
- [ExecutionFeedbackListener Objects](#)
- [ExecutionHandle Objects](#)
- [ExecutionState Objects](#)
- [ExecutionStateEvent Objects](#)
- [ExecutionStateListener Objects](#)
- [Task Objects](#)
- [TaskFactory Objects](#)
- [TaskRunner Objects](#)

Related information

- [ExecutionFeedbackEvent Objects](#)
- [ExecutionFeedbackListener Objects](#)
- [ExecutionHandle Objects](#)
- [ExecutionState Objects](#)
- [ExecutionStateEvent Objects](#)
- [ExecutionStateListener Objects](#)
- [Task Objects](#)
- [TaskFactory Objects](#)
- [TaskRunner Objects](#)

ExecutionFeedbackEvent Objects

Feedback received from an execution task. Feedback is only received from tasks executing a **Node** or **Stream**.

Constants:

- **MESSAGE_ID_EXECUTION_STARTED** (int) : Message ID: execution started.
- **MESSAGE_ID_EXECUTION_STOPPED** (int) : Message ID: execution stopped.
- **MESSAGE_ID_INTERRUPTED** (int) : Message ID: interrupted.
- **MESSAGE_ID_OPTIMIZATION_STARTED** (int) : Message ID: optimization started.
- **MESSAGE_ID_OPTIMIZATION_STOPPED** (int) : Message ID: optimization stopped.
- **MESSAGE_ID_OTHER** (int) : Message ID: other.
- **MESSAGE_ID_PREPARATION_STARTED** (int) : Message ID: preparation started.
- **MESSAGE_ID_PREPARATION_STOPPED** (int) : Message ID: preparation stopped.
- **MESSAGE_ID_PROCESS_EXECUTION_STARTED** (int) : Message ID: external process execution started.
- **MESSAGE_ID_PROCESS_EXECUTION_STOPPED** (int) : Message ID: external process execution stopped.
- **MESSAGE_ID_SQL_EXECUTION_STARTED** (int) : Message ID: SQL execution started.
- **MESSAGE_ID_SQL_EXECUTION_STOPPED** (int) : Message ID: SQL execution stopped.
- **SEVERITY_ERROR** (int) : Severity level: error.
- **SEVERITY_INFORMATION** (int) : Severity level: information.
- **SEVERITY_WARNING** (int) : Severity level: warning.
- **TYPE_DIAGNOSTIC** (int) : Event type: diagnostic message.
- **TYPE_PROGRESS** (int) : Event type: percentage progress.
- **TYPE_RECORD_COUNT** (int) : Event type: record count.

createExecutionFeedbackEvent(handle, type, severity, message) : ExecutionFeedbackEvent

handle (**ExecutionHandle**) : the **ExecutionHandle** which originated the event

type (int) : the type of event

severity (int) : the severity level of the event

message (string) : the message associated with the event

Creates a new execution feedback event. **get messageId()** returns **MESSAGE_ID_OTHER**.

createExecutionFeedbackEvent(handle, type, severity, messageId, message) : ExecutionFeedbackEvent

handle (**ExecutionHandle**) : the **ExecutionHandle** which originated the event

type (int) : the type of event

severity (int) : the severity level of the event

messageId (int) : the message ID associated with the event

message (string) : the message associated with the event

Creates a new execution feedback event. If the value of **messageId** is not recognised, **get messageId()** returns **MESSAGE_ID_OTHER**.

e.getExecutionHandle() : ExecutionHandle

Returns the **ExecutionHandle** that raised the event. This is the source of the event cast to type **ExecutionHandle**.

e.getMessage() : string

Returns the message associated with this event.

For an event of type **TYPE_DIAGNOSTIC** the result is the diagnostic message localized for the session.

For an event of type **TYPE_RECORD_COUNT** the result consists of two decimal integers separated by a space, representing, respectively, the number of records read and written. For example:

```
"15000 0"
```

(15000 records read, none written).

For an event of type **TYPE_PROGRESS** the result consists of a single floating point number in the range

```
0.0 <= n <= 100.0
```

representing (an estimate of) the percentage of work completed. For example:

```
"26.3"
```

(26.3% completed).

```
e.getMessageId() : int
```

Returns the message ID associated with this event. This is one of the **MESSAGE_ID** constants.

```
e.getSeverity() : int
```

Returns the severity level of this event. The result is one of the **SEVERITY** constants declared above. The severity levels **SEVERITY_WARNING** and **SEVERITY_ERROR** are associated with the event type **TYPE_DIAGNOSTIC** and indicate a warning or error condition on the server. An error event ultimately causes execution to fail.

```
e.getType() : int
```

Returns the type of feedback represented by this event. The result is one of the **TYPE** constants declared above.

Related information

- [Tasks and execution](#)
-

ExecutionFeedbackListener Objects

Subclass of **EventListener**.

Listener for **ExecutionFeedbackEvent**.

```
e.executionFeedback(event)
event (ExecutionFeedbackEvent) :
```

Called when execution feedback is produced.

Related information

- [Tasks and execution](#)
-

ExecutionHandle Objects

Monitors and controls execution of a **Task** by a **Session**. A new handle is created for each execution.

```
e.getErrorMessag() : string
```

Returns an error message if execution terminated with an error and a message is available; returns **None** otherwise.

```
e.getExecutionState() : ExecutionState
```

Returns the latest **ExecutionState** of the associated task.

```
e.getExitCode() : int
```

Returns the exit code from executing the task. Typically this is either 0 to indicate success or 1 to indicate failure although some tasks may use different conventions. The result of querying the exit code before the task has finished executing is undefined.

```
e.getResult() : Object
```

Returns the result of the task if execution terminated with success and the task produced a result. Returns **None** if the task is still executing, or if it terminated in a state other than **SUCCESS**, or if it did not return a result.

```
e.getTask() : Task
```

Returns the **Task** being executed.

```
e.terminate() : ExecutionState
```

Terminates execution of the associated task and returns its final **ExecutionState**. The call blocks until the task has finished. The return value may not be **TERMINATED** if execution had completed before the termination request was sent.

```
e.terminate(milliseconds) : ExecutionState
```

```
milliseconds (long) : the maximum time to wait for the task to complete
```

Terminates execution of the associated task, waits until the task has finished or until the specified timeout has expired (whichever is sooner) and returns the execution state of the task. The result may not be **TERMINATED** if execution had completed before the termination request was sent or if execution had not completed before the timeout expired. A timeout of 0 or less means to wait forever.

```
e.waitForCompletion() : ExecutionState
```

Waits until the associated task has finished executing and returns its final **ExecutionState**.

```
e.waitForCompletion(milliseconds) : ExecutionState
```

```
milliseconds (long) : the maximum time to wait for completion
```

Waits until the associated task has finished executing, or until the specified timeout has expired (whichever is sooner) and returns the **ExecutionState** of the task. A timeout of 0 or less means to wait forever.

Related information

- [Tasks and execution](#)
-

ExecutionState Objects

Subclass of **Enum**.

This class enumerates the task execution states.

Constants:

- **ERROR (ExecutionState)** : The task has completed with an error.
- **EXECUTING (ExecutionState)** : The task is still executing.
- **SUBMITTED (ExecutionState)** : The task has been submitted for execution but not yet started.
- **SUCCESS (ExecutionState)** : The task has completed successfully.
- **TERMINATED (ExecutionState)** : The task has completed prematurely as the result of a terminate request.
- **TERMINATING (ExecutionState)** : A terminate request has been issued but the task has not yet completed.

```
getEnum(name) : ExecutionState
```

```
name (string) : the enumeration name
```

Returns the enumeration with the supplied name or **None** if no enumeration exists for the supplied name.

```
getValues() : ExecutionState[]
```

Returns an array containing all the valid values for this enumeration class.

```
e.isCompletedState() : boolean
```

Returns **True** if this state represents a task that is no longer executing. This true for **SUCCESS**, **TERMINATED** and **ERROR** states.

Related information

- [Tasks and execution](#)
-

ExecutionStateEvent Objects

Indicates a state change in a task. An event is fired when a task moves into a new state, and the new state is obtained from the event by calling `getExecutionState`.

```
createExecutionStateEvent(handle, state) : ExecutionStateEvent
```

`handle (ExecutionHandle)` : the `ExecutionHandle` for the event

`state (ExecutionState)` : the new execution state.

Creates a new execution state event.

```
e.getExecutionHandle() : ExecutionHandle
```

Returns the `ExecutionHandle` that raised the event.

```
e.getExecutionState() : ExecutionState
```

Returns the new execution state which caused this event to be raised. The return value is one of the constants defined by `ExecutionState`. Note: This may not correspond with the state that the task is now in since a subsequent state change may have occurred.

Related information

- [Tasks and execution](#)

ExecutionStateListener Objects

Subclass of `EventListener`.

Listener for `ExecutionStateEvent`.

```
e.executionStateChanged(event)
event (ExecutionStateEvent) :
```

Called when the execution state changes.

Related information

- [Tasks and execution](#)

Task Objects

A `Task` represents an operation that can be performed by the Modeler API.

```
t.getResult() : Object
```

Returns any object produced as a result of executing the task. This may be `None` if the task has not been executed yet, or there was an error during execution or if the task does not produce a result.

Related information

- [Tasks and execution](#)

TaskFactory Objects

The `TaskFactory` is used to create instances of `Task`. A task can be executed synchronously or asynchronously by the `Session`. Session-owned objects that are passed to `TaskFactory` methods must be executed by the session that owns them.

```
t.createDeleteFileTask(filePath) : Task
```

`filePath (string)` :- The path of file on server.

Create a task that delete the file using the specified file path.

```
t.createDownloadFileTask(path, outputStream) : Task
```

```

path (string) : the path of the remote file on server
outputStream (OutputStream) : the output stream

Creates a task that copies the content of a file on server to the specified output stream. The task will fail if the file cannot be read.

t.createDropTableTask(conn, tableName) : Task

conn (ServerDatabaseConnection) : - The ServerDatabaseConnection used for connect to the database.

tableName (string) : - The name of the table which you want to delete.

Create a task that drop the table using the specified table name.

t.createExportDocumentTask(document, outputStream, fileFormat) : Task

document (DocumentOutput) : the DocumentOutput object

outputStream (OutputStream) : the output stream

fileFormat (FileFormat) : the FileFormat to be used

Creates a task that exports a DocumentOutput object to an output stream using the specified FileFormat name. Calling getResult() on the completed task returns None.

Exceptions:

OwnerException : if the output is not owned by the same session that owns the task factory

ExportFormatException : if the document does not support the export format

t.createExportModelTask(model, outputStream, fileFormat) : Task

model (ModelOutput) : the ModelOutput object

outputStream (OutputStream) : the output stream

fileFormat (FileFormat) : the FileFormat to be used

Creates a task that exports a ModelOutput object to an output stream using the specified FileFormat name. Calling getResult() on the completed task returns None.

Exceptions:

OwnerException : if the output is not owned by the same session that owns the task factory

ExportFormatException : if the model does not support the export format

t.createExportModelTask(modelApplier, outputStream, fileFormat) : Task

modelApplier (Node) : the Node object

outputStream (OutputStream) : the output stream

fileFormat (FileFormat) : the FileFormat to be used

Creates a task that exports a Node to an output stream using the specified FileFormat name. Calling getResult() on the completed task returns None.

Exceptions:

OwnerException : if the output is not owned by the same session that owns the task factory

ExportFormatException : if the model does not support the export format

t.createExportStreamTask(stream, outputStream, fileFormat) : Task

stream (Stream) : the stream to be exported

outputStream (OutputStream) : the output stream

fileFormat (FileFormat) : the export file format

Creates a task that export a stream description to an output stream using specified file format.

Exceptions:

OwnerException : if the node is not owned by the same session that owns the task factory

```

ExportFormatException : if the stream does not support the export format

```
t.createImportPMMLModelTask(inputStream) : Task
```

inputStream (`InputStream`) : the input stream

Creates a task that imports a `ModelOutput` object from an input stream. Calling `getResultSet()` on the completed task returns an instance of `ModelOutput`.

```
t.createOpenDocumentTask(inputStream) : Task
```

inputStream (`InputStream`) : the input stream

Creates a task that reads a `DocumentOutput` object from an input stream. Calling `getResultSet()` on the completed task returns an instance of `DocumentOutput`. The document is not added to the output manager.

```
t.createOpenDocumentTask(inputStream, autoManage) : Task
```

inputStream (`InputStream`) : the input stream

autoManage (`boolean`) : whether the document should be added to the output manager

Creates a task that reads a `DocumentOutput` object from an input stream. Calling `getResultSet()` on the completed task returns an instance of `DocumentOutput`. Code that needs to open documents privately without having them made visible to the user should set the `autoManage` flag to `False`.

```
t.createOpenModelTask(inputStream) : Task
```

inputStream (`InputStream`) : the input stream

Creates a task that reads a `ModelOutput` object from an input stream. Calling `getResultSet()` on the completed task returns an instance of `ModelOutput`. The model is not added to the model manager.

```
t.createOpenModelTask(inputStream, autoManage) : Task
```

inputStream (`InputStream`) : the input stream

autoManage (`boolean`) : whether the model should be added to the model manager

Creates a task that reads a `ModelOutput` object from an input stream. Calling `getResultSet()` on the completed task returns an instance of `ModelOutput`. Code that needs to open models privately without having them made visible to the user should set the `autoManage` flag to `False`.

```
t.createOpenProcessorTask(inputStream, stream) : Task
```

inputStream (`InputStream`) : the input stream

stream (`Stream`) : the `Stream`

Creates a task that reads a `Node` object from an input stream and inserts it into the supplied `Stream`. Calling `getResultSet()` on the completed task returns an instance of `Node`.

Exceptions:

OwnerException : if the stream is not owned by the same session that owns the task factory

```
t.createOpenStreamTask(inputStream) : Task
```

inputStream (`InputStream`) : the input stream

Creates a task that reads a `Stream` object from an input stream. Calling `getResultSet()` on the completed task returns an instance of `Stream`.

```
t.createOpenStreamTask(inputStream, autoManage) : Task
```

inputStream (`InputStream`) : the input stream

autoManage (`boolean`) : whether the opened stream should be added to the stream manager

Creates a task that reads a `Stream` object from an input stream. Calling `getResultSet()` on the completed task returns an instance of `Stream`. Code that needs to open streams privately without having them made visible to the user should set the `autoManage` flag to `False`.

```
t.createSaveDocumentTask(document, outputStream) : Task
```

document (`DocumentOutput`) : the `DocumentOutput` object

outputStream (`OutputStream`) : the output stream

Creates a task that saves a `DocumentOutput` object to an output stream. Calling `getResultSet()` on the completed task returns `None`.

Exceptions:

OwnerException : if the document output is not owned by the same session that owns the task factory

t.createSaveModelTask(model, outputStream) : Task

model (ModelOutput) : the **ModelOutput** object

outputStream (OutputStream) : the output stream

Creates a task that saves a **ModelOutput** object to an output stream. Calling **getResult()** on the completed task returns **None**.

Exceptions:

OwnerException : if the model output is not owned by the same session that owns the task factory

t.createSaveProcessorTask(node, outputStream) : Task

node (Node) : the **Node** object

outputStream (OutputStream) : the output stream

Creates a task that saves a **Node** object to an output stream. Calling **getResult()** on the completed task returns **None**.

Exceptions:

OwnerException : if the node is not owned by the same session that owns the task factory

t.createSaveStreamTask(stream, outputStream) : Task

stream (Stream) : the **Stream** object

outputStream (OutputStream) : the output stream

Creates a task that saves a **Stream** object to an output stream. Calling **getResult()** on the completed task returns **None**.

Exceptions:

OwnerException : if the stream is not owned by the same session that owns the task factory

t.createUpdatePMMLModelTask(modelOutput, inputStream) : Task
modelOutput (ModelOutput) :
inputStream (InputStream) :

Creates a task that updates the **ModelOutput** object with PMML read from an input stream. The task will fail if the **ModelOutput** is not based on a PMML model or the **ModelOutput** and the PMML algorithms are not the same.

Exceptions:

OwnerException : if the model output is not owned by the same session that owns the task factory

t.createUploadFileTask(path, inputStream) : Task

path (string) : the path of the remote file on server

inputStream (InputStream) : the input stream

Creates a task that copies the content of the specified input stream to a file on server, replacing any existing file content. The task will fail if the file cannot be written.

Related information

- [Tasks and execution](#)
-

TaskRunner Objects

The **TaskRunner** provides a convenient way of creating and running tasks synchronously.

t.createStream(name, autoConnect, autoManage) : Stream

name (string) : the object's name

autoConnect (boolean) : whether the stream should be auto-connected to the server

`autoManage (boolean)` : whether the stream should be added to the stream manager

Creates and returns a new `Stream`. Note that code that needs to create streams privately without having them made visible to the user should set the `autoManage` flag to `False`.

Exceptions:

`ServerConnectionException` : if the Session is already connected and the auto-connect flag is `True` but a new connection could not be created for the stream

t.exportDocumentToFile(documentOutput, filename, fileFormat)

`documentOutput (DocumentOutput)` : the document to be exported

`filename (string)` : the exported file path

`fileFormat (FileFormat)` : the export file format

Exports the stream description to a file using the specified file format.

Exceptions:

`OwnerException` : if the document is not owned by the task runner session

`SessionException` : if the document cannot be exported for some reason

`ExportFormatException` : if the document does not support the export format

t.exportModelSummaryToFile(modelOutput, filename, fileFormat)

`modelOutput (ModelOutput)` : the model to be exported

`filename (string)` : the exported file path

`fileFormat (FileFormat)` : the export file format which is either `FileFormat.PLAIN_TEXT` or `FileFormat.HTML`.

Exports the model summary to a file using the specified file format. Model summaries can only be exported as text or HTML.

Exceptions:

`OwnerException` : if the model is not owned by the task runner session

`SessionException` : if the model cannot be exported for some reason

`ExportFormatException` : if the model does not support the export format

t.exportModelSummaryToFile(node, filename, fileFormat)

`modelOutput (Node)` : the model applier node to be exported

`filename (string)` : the exported file path

`fileFormat (FileFormat)` : the export file format which is either `FileFormat.PLAIN_TEXT` or `FileFormat.HTML`.

Exports the model summary in the supplied node to a file using the specified file format. Model summaries can only be exported as text or HTML.

Exceptions:

`OwnerException` : if the model is not owned by the task runner session

`SessionException` : if the model cannot be exported for some reason

`ExportFormatException` : if the model does not support the export format

t.exportModelToFile(modelOutput, filename, fileFormat)

`modelOutput (ModelOutput)` : the model to be exported

`filename (string)` : the exported file path

`fileFormat (FileFormat)` : the export file format

Exports the model to a file using the specified file format.

Exceptions:

`OwnerException` : if the model is not owned by the task runner session

`SessionException` : if the model cannot be exported for some reason

`ExportFormatException` : if the model does not support the export format

t.exportModelToFile(node, filename, fileFormat)

`modelOutput (Node)` : the model applier node to be exported

`filename (string)` : the exported file path

`fileFormat (FileFormat)` : the export file format

Exports the model in the supplied node to a file using the specified file format.

Exceptions:

`OwnerException` : if the model is not owned by the task runner session

`SessionException` : if the model cannot be exported for some reason

`ExportFormatException` : if the model does not support the export format

t.exportOutputToFile(object, filename, fileFormat, options)

`object (PropertiedObject)` : the object with the output content

`filename (string)` : the file name where the output should be exported to

`fileFormat (FileFormat)` : the file format

`options (Map)` : either None or a hash table of attribute values

Exports the output content from the supplied object to the specified file and format. The following document formats are supported:

- `FileFormat.PLAIN_TEXT`
- `FileFormat.HTML`
- `FileFormat.RTF`
- `FileFormat.SPV`
- `FileFormat.SPW`
- `FileFormat.SPSS_WEB_REPORT`
- `FileFormat.COGNOS_ACTIVE_REPORT`
- `FileFormat.PDF`
- `FileFormat.MS_EXCEL`
- `FileFormat.MS_EXCEL2007`
- `FileFormat.MS_EXCEL2007_M`
- `FileFormat.MS_POWERPOINT`

The following image formats are supported:

- `FileFormat.BITMAP`
- `FileFormat.PNG`
- `FileFormat.JPEG`
- `FileFormat.TIFF`

If `options` are supplied, the value is a table of attribute/value pairs. For document exports, the options are:

- `alternate_log_file_location` (string): the location of the log file if this is to be expored separately
- `custom_report_title` (string): a custom title for a web report

- `excel_location_option` ("AddColumns", "AddRows", "OverwriteAtCellRef"): where the content should be exported to in Excel
- `excel_operation_option` ("CreateWorkbook", "CreateWorksheet", "ModifyWorksheet"): how the content should be exported to Excel
- `excel_sheetname` (string): the Excel sheet name
- `excel_starting_cell_ref` (string): the Excel starting cell references
- `export_html_headers` (Boolean): whether to export HTML headers
- `export_image_maps` (Boolean): whether to create image maps
- `export_images` (Boolean): whether images should be exported separately for formats that don't support images
- `html_with_style` (Boolean): whether style information should be saved
- `include_footnotes_and_captions` (Boolean): whether footnotes and captions should be included
- `interactive_layer` (Boolean): whether an interactive layer should be included in formats that support them
- `is_logs_excluded` (Boolean): whether exclude logs
- `is_notes_excluded` (Boolean): whether to exclude notes
- `is_text_excluded` (Boolean): whether to exclude text content
- `javascript_file_name` (string): for interactive output, the location of the JavaScript file if separate
- `model_view_option` ("MVExportAll", "MVExportVisible", "MVPrintSetting"): which model views should be exported
- `page_break_between_table` (Boolean): whether to create a page break between tables
- `page_setup` (Map): the page setup settings (see below)
- `pdf_embed_bookmarks` (Boolean): whether to embed bookmarks in PDF document
- `pdf_embed_fonts` (Boolean): whether to embed fonts in the PDF document
- `pivot_table_option` ("PTEExportAllLayers", "PTEExportVisibleLayers", "PTUsePrintLayerSetting"): how pivot tables should be handled
- `ppt_use_viewer_outlines` (Boolean): whether to use viewer outlines in the PowerPoint
- `preserve_break_points` (Boolean): whether to preserve break points
- `table_style` (string): the name of the table style
- `text_encoding` ("utf8", "utf16"): what text encoding to use in text files
- `text_doc_option` ("TxtUseSpaces", "TxtUseTabs"): whether to uses spaces or tabs in text files
- `text_width_autofit` (Boolean): whether to make table columns fit the content in text files
- `txt_column_width` (integer): if table columns are not fitted to their content in text files, defines the column width
- `col_txt_border_character` (character): the table column character when exporting to a text file
- `row_txt_border_character` (character): the table column character when exporting to a text file
- `viewer_doc_name` (string): the name of the document which should be displayed in the viewer
- `wide_tables_option` ("WT_Extend", "WT_Shrink", "WT_Wrap"): defines how wide tables should be handled

For graph exports, the options are:

- `bmp_compress_image` (Boolean): whether to compress bitmap images
- `graph_image_size` (integer): how much to scale the exported image as a percentage of the original
- `graph_invert_colors` (Boolean): whether to invert the colors
- `graph_type` ("BMP", "JPG", "PNG", "TIF"): defines the graph type to be used when exporting images separately from the non-image content
- `model_view_option` ("MVExportAll", "MVExportVisible", "MVPrintSetting"): which model views should be exported
- `png_color_depth` ("BlackWhite", "Grays", "SixteenColors", "TrueColor24", "TrueColor36", "TwoFiftySixColors"): the color depth when exporting to PNG
- `tiff_color_space` ("CMYK", "RGB"): the color space when exporting to TIF
- `tiff_compress_image` (Boolean): whether to compress TIF images

For the `page_setup`, the options are:

- `height` (Double): the page height
- `width` (Double): the page width
- `bottom_margin` (Double): the bottom margin
- `left_margin` (Double): the left margin
- `right_margin` (Double): the right margin
- `top_margin` (Double): the top margin
- `orientation` (Integer): 0 for portrait, 1 for landscape
- `units` ("Centimeters", "Millimeters", "Inches", "PrintPoints"): the units that the margin and dimension values represent

Exceptions:

`OwnerException` : if the object is not owned by the current session

`ExportFormatException` : if the export format is not supported

`SessionException` : if some other error occurs

t.exportStreamToFile(stream, filename, fileFormat)

`stream` (`Stream`) : the stream to be exported

`filename` (string) : the exported file path

fileFormat (FileFormat) : the export file format

Exports the stream description to a file using the specified file format.

Exceptions:

OwnerException : if the stream is not owned by the task runner session

SessionException : if the stream cannot be exported for some reason

ExportFormatException : if the stream does not support the export format

t.insertNodeFromFile(filename, diagram) : Node

filename (string) : the file path

diagram (Diagram) : the diagram that the node should be inserted into

Reads and returns a node from the specified file, inserting it into the supplied diagram. Note that this can be used to read both **Node** and **SuperNode** objects.

Exceptions:

OwnerException : if the diagram is not owned by the task runner session

SessionException : if the node cannot be loaded for some reason

ObjectLockedException : if the stream that the node is being added to is locked

t.openDocumentFromFile(filename, autoManage) : DocumentOutput

filename (string) : the document file path

autoManage (boolean) : whether the document should be added to the output manager

Reads and returns a document from the specified file.

Exceptions:

SessionException : if the document cannot be loaded for some reason

t.openModelFromFile(filename, autoManage) : ModelOutput

filename (string) : the model file path

autoManage (boolean) : whether the model should be added to the model manager

Reads and returns a model from the specified file.

Exceptions:

SessionException : if the model cannot be loaded for some reason

t.openStreamFromFile(filename, autoManage) : Stream

filename (string) : the stream file path

autoManage (boolean) : whether the stream should be added to the stream manager

Reads and returns a stream from the specified file.

Exceptions:

SessionException : if the stream cannot be loaded for some reason

t.saveDocumentToFile(documentOutput, filename)

documentOutput (DocumentOutput) : the document to be saved

filename (string) : the document file path

Saves the document to the specified file location.

Exceptions:

OwnerException : if the document is not owned by the task runner session

SessionException : if the document cannot be saved for some reason

t.saveModelToFile(modelOutput, filename)

modelOutput (ModelOutput) : the model to be saved

filename (string) : the model file path

Saves the model to the specified file location.

Exceptions:

OwnerException : if the model is not owned by the task runner session

SessionException : if the model cannot be saved for some reason

t.saveNodeToFile(node, filename)

node (Node) : the node to be saved

filename (string) : the node file path

Saves the model to the specified file location.

Exceptions:

OwnerException : if the node is not owned by the task runner session

SessionException : if the node cannot be saved for some reason

t.saveStreamToFile(stream, filename)

stream (Stream) : the stream to be saved

filename (string) : the stream file path

Saves the stream to the specified file location.

Exceptions:

OwnerException : if the stream is not owned by the task runner session

SessionException : if the stream cannot be saved for some reason

Streams and SuperNodes

This provides objects that perform data processing and model building.

- [BuiltObject Objects](#)
- [CFNode Objects](#)
- [CompositeModelApplier Objects](#)
- [CompositeModelBuilder Objects](#)
- [CompositeModelOutput Objects](#)
- [CompositeModelOwner Objects](#)
- [CompositeModelResults Objects](#)
- [SuperNode Objects](#)
- [SuperNodeDiagram Objects](#)
- [SuperNodeType Objects](#)
- [DataReader Objects](#)
- [DataTransformer Objects](#)
- [DataWriter Objects](#)
- [DiagramConnector Objects](#)
- [DocumentBuilder Objects](#)
- [DocumentOutput Objects](#)

- [DocumentOutputType Objects](#)
- [ExportFormatException Objects](#)
- [GraphBuilder Objects](#)
- [GraphOutput Objects](#)
- [InitialNode Objects](#)
- [ProcessNode Objects](#)
- [InvalidEditException Objects](#)
- [ModelApplier Objects](#)
- [ModelBuilder Objects](#)
- [ModelOutput Objects](#)
- [ModelOutputType Objects](#)
- [ObjectBuilder Objects](#)
- [Node Objects](#)
- [Diagram Objects](#)
- [NodeFilter Objects](#)
- [Stream Objects](#)
- [PublishedImage Objects](#)
- [ReportBuilder Objects](#)
- [ReportOutput Objects](#)
- [RowSetBuilder Objects](#)
- [RowSetOutput Objects](#)
- [TerminalNode Objects](#)
- [Updatable Objects](#)
- [Updater Objects](#)

Related information

- [BuiltObject Objects](#)
- [CFNode Objects](#)
- [CompositeModelApplier Objects](#)
- [CompositeModelBuilder Objects](#)
- [CompositeModelOutput Objects](#)
- [CompositeModelOwner Objects](#)
- [CompositeModelResults Objects](#)
- [SuperNode Objects](#)
- [SuperNodeDiagram Objects](#)
- [SuperNodeType Objects](#)
- [DataReader Objects](#)
- [DataTransformer Objects](#)
- [DataWriter Objects](#)
- [DiagramConnector Objects](#)
- [DocumentBuilder Objects](#)
- [DocumentOutput Objects](#)
- [DocumentOutputType Objects](#)
- [ExportFormatException Objects](#)
- [GraphBuilder Objects](#)
- [GraphOutput Objects](#)
- [InitialNode Objects](#)
- [ProcessNode Objects](#)
- [InvalidEditException Objects](#)
- [ModelApplier Objects](#)
- [ModelBuilder Objects](#)
- [ModelOutput Objects](#)
- [ModelOutputType Objects](#)
- [ObjectBuilder Objects](#)
- [Node Objects](#)
- [Diagram Objects](#)
- [NodeFilter Objects](#)
- [Stream Objects](#)
- [PublishedImage Objects](#)
- [ReportBuilder Objects](#)
- [ReportOutput Objects](#)
- [RowSetBuilder Objects](#)
- [RowSetOutput Objects](#)
- [TerminalNode Objects](#)
- [Updatable Objects](#)
- [Updater Objects](#)

BuiltObject Objects

Subclass of **PropertiedObject**.

This encapsulates the concrete results of executing **objectBuilder** nodes.

b.close()

Closes this built object and releases its external resources. Closing an object which is already closed has no effect, otherwise the behaviour of a method applied to a closed object is undefined except where stated.

b.getBuilderProcessorID() : string

Returns the ID of the object builder node that built this object. The empty string is returned if this information is not available.

b.getBuilderStreamID() : string

Returns the temporary session ID of the stream containing the object builder node that built this object. If the stream is closed and subsequently re-opened, the re-opened stream's ID will be different from the value returned here.

The empty string is returned if this information is not available.

b.getID() : string

Returns the temporary session ID of this object. A new ID is allocated each time a new built object is created or opened and the ID is not persisted. This means that if the same object is re-opened multiple times, each object will include a different ID.

b.isClosed() : boolean

Returns **True** if this object has been closed.

b.isExportable(fileFormat) : boolean

fileFormat (FileFormat) : the FileFormat

Returns **True** if this object can be exported using the supplied **FileFormat** or **False** otherwise.

Related information

- [Streams and SuperNodes](#)

CFNode Objects

Subclass of **Node**.

This identifies **Node** objects that are implemented via the Component Framework.

c.getFeature() : Object

Return the underlying CF feature of the CF node.

Related information

- [Streams and SuperNodes](#)

CompositeModelApplier Objects

Subclass of **ModelApplier**, **CompositeModelOwner**.

This encapsulates the auto-model applier nodes **ModelApplier** that can ensemble the scores from multiple models into a single score.

Related information

- [Streams and SuperNodes](#)

CompositeModelBuilder Objects

Subclass of `ModelBuilder`.

This encapsulates the auto-model builder nodes `ModelBuilder` that can build and evaluate multiple models using different modeling algorithms and settings.

```
c.getAllModelAlgorithms() : List
```

Returns the model builder node ids that are available in this `CompositeModelBuilder`.

```
c.isAlgorithmEnabled(modelBuilderId) : boolean
```

```
modelBuilderId (string) : build model id
```

Returns whether the supplied algorithm is enabled. If the supplied algorithm ID is not one of the algorithms available for this model builder, the method returns `False`. An algorithm may be enabled in the model builder, but invalid due to other settings (see `isAlgorithmInvalid(String)`).

```
c.isAlgorithmInvalid(algorithmID) : boolean
```

```
algorithmID (string) :
```

Returns whether the supplied algorithm is invalid. An algorithm may be enabled (see `isAlgorithmEnabled(String)`) in this model builder, but be invalid due to other settings. For instance, the target measure may not be valid for the algorithm. If the algorithm is invalidated in this way it will not be built, even if it is enabled.

```
c.setAlgorithmEnabled(modelBuilderId, value)
```

```
modelBuilderId (string) : build model id
```

```
value (boolean) : the enabled value
```

Sets whether the supplied algorithm is enabled. If the supplied algorithm ID is not one of the algorithms available for this model builder, the method call will have no effect.

Related information

- [Streams and SuperNodes](#)

CompositeModelOutput Objects

Subclass of `ModelOutput`, `CompositeModelOwner`.

This encapsulates the concrete results of executing `ModelBuilder` nodes or by opening or importing Composite models. This will be the results of the auto models.

```
c.getModelDetail() : CompositeModelDetail
```

Returns the underlying composite model representation.

Related information

- [Streams and SuperNodes](#)

CompositeModelOwner Objects

This encapsulates objects that own auto-built models. These models can be ensembled and the scores from multiple models combined into a single score.

```
c.getCompositeModelDetail() : CompositeModelDetail
```

Returns the underlying composite model representation.

```
c.getModelResult(resultId) : CompositeModelResult
```

```
resultId (string) :
```

Returns the model results for the specified ID or `None` if no such model result exists.

```
c.getModelResultIDs() : List
```

Returns the list of model result IDs. The result ID is the same as the name of the individual composite model result.

```
c.isModelResultUsed(resultId) : boolean  
resultId (string) :
```

Returns whether the specified model result ID is enabled/active.

```
c.removeModelResults(resultIds)
```

```
resultIds (List) : the list of result IDs to be removed
```

Permanently removes the supplied model results from this composite model object owner.

```
c.setModelResultUsed(resultId, value)
```

```
resultId (string) : the result ID
```

```
value (boolean) : True if the model should be enabled, False if not
```

Sets whether the specified model result ID is active. Note that supplying model results that have previously been removed will have no effect i.e. they will not be re-added into the composite model object owner.

Related information

- [Streams and SuperNodes](#)
-

CompositeModelResults Objects

This encapsulates the concrete results for the individual models forming the Composite Model.

```
c.getModelMeasureLabel(measure) : string
```

```
measure (string) : the evaluation measure name
```

Returns a descriptive label for the specified evaluation measure or `None` if the measure is not valid for this result.

```
c.getModelMeasurePartitionCount() : int
```

Returns the number of partitions for which evaluation measures exist. Partitions (where they exist) are numbered: 0=Training, 1=Testing, 2=Validation.

```
c.getModelMeasureType(measure) : StorageType
```

```
measure (string) : the evaluation measure name
```

Returns the data type of the specified evaluation measure or `None` if the measure is not valid for this result.

```
c.getModelMeasureValue(measure, partition) : Object
```

```
measure (string) : the evaluation measure name
```

```
partition (int) : the partition index
```

Returns the value of the specified evaluation measure in the specified partition, or `None` if either the measure or the partition is not valid for this result.

```
c.getModelMeasures() : List
```

Returns the list of measures used to evaluate this model result.

```
c.getModelName() : string
```

Returns the name of this model result. The model name is expected to be unique within a set of composite model result objects.

```
c.getModelOutput() : ModelOutput
```

Returns the model output of the composite model result.

```
c.isUsed() : boolean
```

Returns `True` if this composite model result is used.

Related information

- [Streams and SuperNodes](#)
-

SuperNode Objects

Subclass of **Node**, **ParameterProvider**.

This encapsulates the functionality of any SuperNode whose behaviour is determined by its constituent nodes.

s.getChildDiagram() : SuperNodeDiagram

Returns the diagram containing the nodes encapsulated by this SuperNode.

s.getCompositeProcessorType() : SuperNodeType

Returns the type of this SuperNode.

s.isPasswordNeeded() : boolean

If the SuperNode is not locked or a password has been entered then return **False**, otherwise return **True**.

s.setCompositeProcessorType(type)

type (SuperNodeType) : the SuperNode type

Sets the type of this SuperNode.

Exceptions:

ObjectLockedException : if the diagram is currently locked

s.verifyPassword(password) : boolean

password (string) : the password to be tested

Check whether the supplied password is valid or not.

Related information

- [Streams and SuperNodes](#)
-

SuperNodeDiagram Objects

Subclass of **Diagram**.

This is the diagram used by **SuperNode** objects to store nodes. It allows connections between the two pseudo nodes, (the input connector and output connector) and other nodes in the diagram to be edited.

s.disconnectInputConnector()

Removes any direct links between the diagram's input connector and other nodes in the diagram. If the SuperNode is an initial node and therefore doesn't have an input connector, this method does nothing.

Exceptions:

ObjectLockedException : if the diagram is currently locked

s.disconnectOutputConnector()

Removes any direct links between the diagram's output connector and other nodes in the diagram. If the SuperNode is an terminal node and therefore doesn't have an input connector, this method does nothing.

Exceptions:

ObjectLockedException : if the diagram is currently locked

s.getInputConnector() : Node

Returns the input connector from this diagram, or **None** if the diagram belongs to an initial node and has no input connector.

```
s.getOutputConnector() : Node
```

Returns the output connector from this diagram, or **None** if the diagram belongs to a terminal node and has no output connector.

```
s.linkFromInputConnector(node)
node (Node) :
```

Creates a link from the input connector to the supplied node.

Exceptions:

OwnerException : if any objects in the path are not owned by this diagram

ObjectLockedException : if the diagram is currently locked

InvalidEditException : if the connection would be invalid

```
s.linkToOutputConnector(node)
node (Node) :
```

Creates a link from the supplied node to the output connector.

Exceptions:

OwnerException : if any objects in the path are not owned by this diagram

ObjectLockedException : if the diagram is currently locked

InvalidEditException : if the connection would be invalid

```
s.runAll(results) : ExecutionHandle
```

results (Collection) : an empty collection that will contain any built objects once execution has completed

Executes the executable nodes within this sub-stream synchronously and waits for it to complete. Returns an **ExecutionHandle** which can be used to access the exit status and any result from the task.

Note that this should only be called on a terminal **SuperNode**.

Exceptions:

OwnerException : if the stream was not created by this session

ObjectLockedException : if the stream is locked

ServerConnectionException : if the stream is not connected to a server

SessionException : if the session is already running another task, cannot execute the task or if execution completes in a state other than **SUCCESS**

```
s.runSelected(nodes, results) : ExecutionHandle
```

nodes (Node[]) : the array of **Node** objects to be executed

results (Collection) : an empty collection that will contain any built objects once execution has completed

Executes the supplied array of nodes synchronously and waits for them to complete. There must be at least one node in the array. Returns an **ExecutionHandle** which can be used to access the exit status and any result from the task.

Note that this should only be called on a terminal **SuperNode**.

Exceptions:

OwnerException : if the nodes are not all owned by this stream

ObjectLockedException : if the stream is locked

ServerConnectionException : if the stream is not connected to a server

SessionException : if the session is already running a stream, another task, cannot execute the task or if execution completes in a state other than **SUCCESS**

IllegalArgumentException : if the array is empty

```
s.unlinkFromInputConnector(node)
node (Node) :
```

Removes any direct link from the input connector to the supplied node.

Exceptions:

`OwnerException` : if any objects in the path are not owned by this diagram

`ObjectLockedException` : if the diagram is currently locked

`InvalidEditException` : if the connection would be invalid

```
s.unlinkToOutputConnector(node)
node (Node) :
```

Removes any direct link from the supplied node to the output connector.

Exceptions:

`OwnerException` : if any objects in the path are not owned by this diagram

`ObjectLockedException` : if the diagram is currently locked

`InvalidEditException` : if the connection would be invalid

Related information

- [Streams and SuperNodes](#)
-

SuperNodeType Objects

Subclass of `Enum`.

This class enumerates the types of SuperNodes.

Constants:

- `INITIAL (SuperNodeType)` : Represents a `SuperNode` that begins a sequence of nodes.
- `PROCESS (SuperNodeType)` : Represents a `SuperNode` that allows data to flow through it.
- `TERMINAL (SuperNodeType)` : Represents a `SuperNode` that ends a sequence of nodes.

```
getEnum(name) : SuperNodeType
```

`name (string)` : the enumeration name

Returns the enumeration with the supplied name or `None` if no enumeration exists for the supplied name.

```
getValues() : SuperNodeType[]
```

Returns an array containing all the valid values for this enumeration class.

Related information

- [Streams and SuperNodes](#)
-

DataReader Objects

Subclass of `InitialNode`.

This encapsulates the functionality of a data node that reads records e.g. from a file or database table.

Related information

- [Streams and SuperNodes](#)
-

DataTransformer Objects

Subclass of **ProcessNode**.

This encapsulates the functionality of a standard inline data node.

Related information

- [Streams and SuperNodes](#)
-

DataWriter Objects

Subclass of **TerminalNode**.

This encapsulates the functionality of a node that writes records e.g. to a file or database table.

```
d.getExportDataModel() : DataModel
```

Returns the **DataModel** being exported by this node.

Related information

- [Streams and SuperNodes](#)
-

DiagramConnector Objects

Subclass of **ProcessNode**.

A node which provides a connection point between a **SuperNodeDiagram** and its containing diagram. A connector routes data between diagrams.

Related information

- [Streams and SuperNodes](#)
-

DocumentBuilder Objects

Subclass of **ObjectBuilder**.

This encapsulates the functionality of a node that builds a viewable output object such as a table, graph or report.

Related information

- [Streams and SuperNodes](#)
-

DocumentOutput Objects

Subclass of **BuiltObject**.

This encapsulates the concrete results of executing **DocumentBuilder** nodes that produce generic document-type outputs.

```
d.getDocumentOutputType() : DocumentOutputType
```

Returns the **DocumentOutputType** of this output. The returned value corresponds with one of the output-type constants defined by **DocumentOutputType**.

```
d.getTypeName() : string
```

Returns the name of this type of output object.

Related information

- [Streams and SuperNodes](#)
-

DocumentOutputType Objects

Subclass of **Enum**.

This class enumerates the types of document outputs that are generated by the different **NodeType** that are **DocumentBuilders** or which can be imported.

Constants:

- **ANALYSIS** (**DocumentOutputType**) :
- **COLLECTION** (**DocumentOutputType**) :
- **CUSTOM_TABLE** (**DocumentOutputType**) :
- **DATA_AUDIT** (**DocumentOutputType**) :
- **DISTRIBUTION** (**DocumentOutputType**) :
- **ENSEMBLE** (**DocumentOutputType**) :
- **EVALUATION** (**DocumentOutputType**) :
- **GRAPH_BOARD** (**DocumentOutputType**) :
- **HISTOGRAM** (**DocumentOutputType**) :
- **MATRIX** (**DocumentOutputType**) :
- **MEANS** (**DocumentOutputType**) :
- **MULTI_PLOT** (**DocumentOutputType**) :
- **QUALITY** (**DocumentOutputType**) :
- **REPORT** (**DocumentOutputType**) :
- **REPORT_DOCUMENT** (**DocumentOutputType**) :
- **SCATTER_PLOT** (**DocumentOutputType**) :
- **SPSS_PROCEDURE** (**DocumentOutputType**) :
- **STATISTICS** (**DocumentOutputType**) :
- **TABLE** (**DocumentOutputType**) :
- **TIME_PLOT** (**DocumentOutputType**) :
- **TRANSFORM** (**DocumentOutputType**) :
- **UNKNOWN** (**DocumentOutputType**) : Represents a document output object whose type cannot be determined.
- **WEB** (**DocumentOutputType**) :

addDocumentOutputType(identifier) : DocumentOutputType
identifier (string) :

Adds a the type with the supplied name. Returns the new type or **None** if a type with the supplied name already exists. This method is for system use only and is only public as an implementation detail.

getEnum(name) : DocumentOutputType

name (string) : the enumeration name

Returns the enumeration with the supplied name or **None** if no enumeration exists for the supplied name.

getValues() : DocumentOutputType[]

Returns an array containing all the valid values for this enumeration class.

d.isUnknown() : boolean

Returns **True** if this value is **UNKNOWN**.

removeDocumentOutputType(identifier)

identifier (string) : the type to be removed

Removes the specified type. The type should have been created with **addDocumentOutputType()**. This method is for system use only and is only public as an implementation detail.

Related information

- [Streams and SuperNodes](#)

ExportFormatException Objects

Subclass of **ModelerException**.

Indicates that an attempt was made to export a built object using an unsupported file format.

No message string is set for this exception.

```
e.getFileFormat() : FileFormat
```

Returns the unsupported export format.

```
e.getObject() : BuiltObject
```

Returns the built object.

Related information

- [Streams and SuperNodes](#)

GraphBuilder Objects

Subclass of **DocumentBuilder**.

This encapsulates the functionality of a node that builds a graph output.

```
g.getGraphOutput() : GraphOutput
```

Returns the most recent **GraphOutput** produced by this node, or None if this **GraphBuilder** has either not been executed before or failed to produce an object during the most recent execution.

Related information

- [Streams and SuperNodes](#)

GraphOutput Objects

Subclass of **DocumentOutput**.

This encapsulates the concrete results of executing **DocumentBuilder** nodes that produce objects that are based on graphs.

Related information

- [Streams and SuperNodes](#)

InitialNode Objects

Subclass of **Node**.

This encapsulates the functionality of any node that begins a sequence of nodes. These typically import data from a data source such as a database or file.

Related information

- [Streams and SuperNodes](#)

ProcessNode Objects

Subclass of **Node**.

This encapsulates the functionality of any node that allows data to flow through it i.e., a **Node** that is neither a **InitialNode** nor a **TerminalNode**.

Related information

- [Streams and SuperNodes](#)
-

InvalidEditException Objects

Subclass of **ModelerException**.

An exception thrown when an invalid attempt is made to edit the links between two or more nodes.

No message string is set for this exception.

`i.getProcessorDiagram() : Diagram`

Returns the **Diagram** that generated the exception.

`i.getProcessorStream() : Stream`

Returns the **Stream** that generated the exception.

Related information

- [Streams and SuperNodes](#)
-

ModelApplier Objects

Subclass of **ProcessNode**.

This encapsulates the functionality of an inline node that applies models.

`m.getBuilderProcessorID() : string`

Returns the ID of the model builder node that built the underlying model. The empty string is returned if this information is not available.

`m.getBuilderStreamID() : string`

Returns the ID of the stream of the model builder node that built the underlying model. The empty string is returned if this information is not available.

`m.getLocalizedName() : string`

Returns the localized algorithm name.

`m.getModelDetail() : ModelDetail`

Returns the underlying model representation.

Related information

- [Streams and SuperNodes](#)
-

ModelBuilder Objects

Subclass of **ObjectBuilder**.

This encapsulates the functionality of a node that builds a model output object.

```
m.getModelOutputType() : ModelOutputType
```

Returns the type of `ModelOutput` that is produced by this type of model builder. The returned value corresponds to one of the model-type constants defined by `ModelOutputType`.

Related information

- [Streams and SuperNodes](#)

ModelOutput Objects

Subclass of `BuiltObject`.

This encapsulates the concrete results of executing `ModelBuilder` nodes or by opening or importing models.

```
m.getLocalizedAlgorithmName() : string
```

Returns the localized algorithm name.

```
m.getModelApplierID() : string
```

Returns the ID of `ModelApplier` that corresponds with this model output or `None` if this model output cannot be applied directly to data.

```
m.getModelDetail() : ModelDetail
```

Returns the underlying model representation.

```
m.getModelOutputType() : ModelOutputType
```

Returns the `ModelOutputType` of this model output. The returned value corresponds with one of the model-type constants defined by `ModelOutputType`.

```
m.getTypeName() : string
```

Returns the name of this type of output object.

Related information

- [Streams and SuperNodes](#)

ModelOutputType Objects

Subclass of `Enum`.

This class enumerates the types of model outputs that are generated by the different `NodeType` that are `ModelBuilders` or which can be imported.

Constants:

- `ANOMALY_DETECTION` (`ModelOutputType`) :
- `APRIORI` (`ModelOutputType`) :
- `ASSOCIATION` (`ModelOutputType`) :
- `AUTO_CLUSTER` (`ModelOutputType`) :
- `BACK_PROPAGATION` (`ModelOutputType`) :
- `BINARY_CLASSIFIER` (`ModelOutputType`) :
- `C50` (`ModelOutputType`) :
- `C5_SPLIT` (`ModelOutputType`) :
- `CARMA` (`ModelOutputType`) :
- `CART` (`ModelOutputType`) :
- `CART_SPLIT` (`ModelOutputType`) :
- `CATEGORIZE` (`ModelOutputType`) :
- `CHAID` (`ModelOutputType`) :
- `CHAID_SPLIT` (`ModelOutputType`) :
- `DB2IM_ASSOCIATION` (`ModelOutputType`) :
- `DB2IM_CLUSTER` (`ModelOutputType`) :

- DB2IM_LOGISTIC (ModelOutputType) :
- DB2IM_NAIVE_BAYES (ModelOutputType) :
- DB2IM_REGRESSION (ModelOutputType) :
- DB2IM_SEQUENCE (ModelOutputType) :
- DB2IM_TIME_SERIES (ModelOutputType) :
- DB2IM_TREE (ModelOutputType) :
- DECISION_LIST (ModelOutputType) :
- DECISION_LIST_SPLIT (ModelOutputType) :
- FACTOR (ModelOutputType) :
- FEATURE_SELECTION (ModelOutputType) :
- KMEANS (ModelOutputType) :
- KOHONEN (ModelOutputType) :
- LINEAR_REGRESSION (ModelOutputType) :
- LINEAR_REGRESSION_SPLIT (ModelOutputType) :
- LOGISTIC_REGRESSION (ModelOutputType) :
- LOGISTIC_SPLIT (ModelOutputType) :
- MS_ASSOCIATION (ModelOutputType) :
- MS_CLUSTERING (ModelOutputType) :
- MS_LOGISTIC (ModelOutputType) :
- MS_NAIVE_BAYES (ModelOutputType) :
- MS_NEURAL_NETWORK (ModelOutputType) :
- MS_REGRESSION (ModelOutputType) :
- MS_SEQUENCE_CLUSTERING (ModelOutputType) :
- MS_TIME_SERIES (ModelOutputType) :
- MS_TREE (ModelOutputType) :
- NET_SPLIT (ModelOutputType) :
- NUMERIC_PREDICTOR (ModelOutputType) :
- ORACLE_ADAPTIVE_BAYES (ModelOutputType) :
- ORACLE_AI (ModelOutputType) :
- ORACLE_DECISION_TREE (ModelOutputType) :
- ORACLE_GLM (ModelOutputType) :
- ORACLE_KMEANS (ModelOutputType) :
- ORACLE_NAIVE_BAYES (ModelOutputType) :
- ORACLE_NMF (ModelOutputType) :
- ORACLE_OCLUSTER (ModelOutputType) :
- ORACLE_SVM (ModelOutputType) :
- QUEST (ModelOutputType) :
- QUEST_SPLIT (ModelOutputType) :
- RULE (ModelOutputType) :
- SEQUENCE (ModelOutputType) :
- SPSS_MODEL (ModelOutputType) :
- TEXT_EXTRACTION (ModelOutputType) :
- TIME_SERIES (ModelOutputType) :
- TWOSTEP (ModelOutputType) :
- UNKNOWN (ModelOutputType) : Represents a model output object whose type cannot be determined.

addModelOutputType(identifier) : ModelOutputType

`identifier (string) :`

Adds a type with the supplied name. Returns the new type or `None` if a type with the supplied name already exists. This method is for system use only and is only public as an implementation detail.

getEnum(name) : ModelOutputType

`name (string) : the enumeration name`

Returns the enumeration with the supplied name or `None` if no enumeration exists for the supplied name.

getValues() : ModelOutputType[]

Returns an array containing all the valid values for this enumeration class.

m.isUnknown() : boolean

Returns `True` if this value is `UNKNOWN`.

removeModelOutputType(identifier)

`identifier (string) : the type to be removed.`

Removes the specified type. The type should have been created with `addModelOutputType()`. This method is for system use only and is only public as an implementation detail.

Related information

- [Streams and SuperNodes](#)

ObjectBuilder Objects

Subclass of `TerminalNode`.

This encapsulates the functionality of a node that builds an output object such as a model or graph.

`o.getBuiltObject() : BuiltObject`

Returns the most recent `BuiltObject` produced by this node, or `None` if this `ObjectBuilder` has either not been executed before or failed to produce an object during the most recent execution.

Related information

- [Streams and SuperNodes](#)

Node Objects

Subclass of `PropertiedObject`, `ContentContainerProvider`.

This is the base interface for all nodes.

`n.flushCache()`

Flushes the cache of this object. Has no effect if the cache is not enabled or is not full.

`n.getID() : string`

Returns the ID of this object. A new ID is created each time a new node is created. The ID is persisted with the node when it is saved as part of a stream so that when the stream is opened, the node IDs are preserved. However, if a saved node is inserted into a stream, the insert node is considered to be a new object and will be allocated a new ID.

`n.getInputDataModel() : DataModel`

Returns the `DataModel` coming into this node.

`n.getOutputDataModel() : DataModel`

Returns the `DataModel` output by this node.

`n.getProcessorDiagram() : Diagram`

Returns the `Diagram` that owns this node.

`n.getProcessorStream() : Stream`

Returns the `Stream` that owns this node.

`n.getTypeName() : string`

Returns the name of this type of node.

`n.getXPosition() : int`

Returns the x position offset of the node in the `Diagram`.

`n.getYPosition() : int`

Returns the y position offset of the node in the `Diagram`.

```

n.isCacheEnabled() : boolean

```

Returns **True** if the cache is enabled, **False** otherwise.

```

n.isCacheFull() : boolean

```

Returns **True** if the cache is full, **False** otherwise.

```

n.isInitial() : boolean

```

Returns **True** if this is an initial node i.e. one that occurs at the start of a stream.

```

n.isInline() : boolean

```

Returns **True** if this is an in-line node i.e. one that occurs mid-stream.

```

n.isTerminal() : boolean

```

Returns **True** if this is a terminal node i.e. one that occurs at the end of a stream.

```

n.run(results) : ExecutionHandle
results (Collection) :

```

Executes this node synchronously and waits for execution to complete. Returns an **ExecutionHandle** which can be used to access the exit status and any result from the task. Equivalent to calling:

```

node.getProcessorStream().runSelected(new Processor[] {node}, results);

```

Exceptions:

- OwnerException** : if there is inconsistent ownership
- ObjectLockedException** : if the owner stream is locked for some reason (for example, it is already executing)
- ServerConnectionException** : if the connection to the server cannot be established
- SessionException** : if some other exception occurs

```

n.setCacheEnabled(val)
val (boolean) :

```

Enables or disables the cache for this object. If the cache is full and the caching becomes disabled, the cache is flushed.

```

n.setPositionBetween(source, target)

```

source (Node) : the predecessor

target (Node) : the successor

Sets the position of the node in the **Diagram** so it is positioned between the supplied nodes.

```

n.setXYPosition(x, y)

```

x (int) : the x offset

y (int) : the y offset

Sets the position of the node in the **Diagram**.

Related information

- [Streams and SuperNodes](#)
-

Diagram Objects

This is the container used to assemble **Node** objects into a connected flow or well-formed sequence of nodes.

```
d.clear()
```

Deletes all nodes from this diagram.

Exceptions:

```
ObjectLockedException : if the diagram is currently locked
```

`d.create(nodeType, name) : Node`

`nodeType (string) : the node type name`

`name (string) : the object's name`

Creates a `Node` of the specified type and adds it to this diagram.

Exceptions:

`ObjectCreationException : if the node cannot be created for some reason`

`ObjectLockedException : if the node cannot be added to the stream`

`d.createAt(nodeType, name, x, y) : Node`

`nodeType (string) : the node type name`

`name (string) : the object's name`

`x (int) : the x location`

`y (int) : the y location`

Creates a `Node` of the specified type and adds it to this diagram at the specified location. If either `x < 0` or `y < 0`, the location is not set.

Exceptions:

`ObjectCreationException : if the node cannot be created for some reason`

`ObjectLockedException : if the node cannot be added to the stream`

`d.createModelApplier(modelOutput, name) : Node`

`modelOutput (ModelOutput) : the model output object`

`name (string) : the new object's name`

Creates a `ModelApplier` derived from the supplied model output object.

Exceptions:

`ObjectCreationException : if the node cannot be created for some reason`

`ObjectLockedException : if the node cannot be added to the stream`

`d.delete(node)`

`node (Node) : the node to be removed`

Deletes the specified node from this diagram. The node must be owned by this diagram.

Exceptions:

`OwnerException : if the node is not owned by this diagram`

`ObjectLockedException : if the diagram is currently locked`

`d.deleteAll(nodes)`

`nodes (Collection) : the collection of nodes to be removed`

Deletes all the specified nodes from this diagram. All nodes in the collection must belong to this diagram.

Exceptions:

`OwnerException : if any node parameters are not owned by this diagram`

`ObjectLockedException : if the diagram is currently locked`

`ClassCastException : if any item in the collection is not of type Node`

`d.disconnect(node)`

`node (Node) : the node to be disconnected`

Removes any links between the supplied node and any other nodes in this diagram.

Exceptions:

OwnerException : if any objects in the path are not owned by this diagram

ObjectLockedException : if the diagram is currently locked

d.findAll(type, label) : Collection

type (string) : the node type

label (string) : the node label

Returns a list of all nodes with the specified type and name. Either the type or name may be **None** in which case the other parameter is used.

d.findAll(filter, recursive) : Collection

filter (NodeFilter) : the node filter

recursive (boolean) : if **True** then composite nodes should be recursively searched

Returns a collection of all nodes accepted by the specified filter. If the recursive flag is **True** then any SuperNodes within this diagram are also searched.

d.findById(id) : Node

id (string) : the node ID

Returns the node with the supplied ID or **None** if no such node exists. The search is limited to the current diagram.

d.findByType(type, label) : Node

type (string) : the node type

label (string) : the node label

Returns the node with the supplied type and/or label. Either the type or name may be **None** in which case the other parameter is used. If multiple nodes match then an arbitrary one is chosen and returned. If no nodes match then the return value is **None**.

d.findDownstream(fromNodes) : List

fromNodes (List) : the start point of the search

Searches from the supplied list of nodes and returns the set of nodes downstream of the supplied nodes. The returned list includes the originally supplied nodes.

d.findUpstream(fromNodes) : List

fromNodes (List) : the start point of the search

Searches from the supplied list of nodes and returns the set of nodes upstream of the supplied nodes. The returned list includes the originally supplied nodes.

d.flushCaches()

Flushes the caches of any cache-enabled **Node** objects in the diagram. Has no effect if caches are not enabled or are not full.

d.insert(source, nodes, newIDs) : List

source (Diagram) : the diagram that owns the nodes to be inserted

nodes (List) : the nodes to be copied

newIDs (boolean) : **True** if new IDs should be generated for each node or **False** if the existing IDs should be re-used

Inserts copies of the nodes in the supplied list. It is assumed that all nodes in the supplied list are contained within the specified diagram. The new IDs flag indicates whether new IDs should be generated for each node, or whether the existing ID should be copied across. It is assumed that all nodes in a diagram have a unique ID so this flag must be set **True** if the source diagram is the same as this diagram. The method returns the list of newly inserted nodes where the order of the nodes is undefined (i.e. the ordering is not necessarily the same as the order of nodes in the input list).

Exceptions:

ObjectLockedException : if the diagram has been locked because of another operation

InvalidEditException : if the edit would be invalid

d.isEnabled(node) : boolean

node (Node) : the node

Returns **True** if the supplied node is enabled.

Exceptions:

OwnerException : if the node is not owned by this diagram

d.isOwner(node) : boolean

node (Node) : the node

Returns **True** if the node is owned by this diagram.

d.isValidLink(source, target) : boolean

source (Node) : the source node

target (Node) : the target node

Returns **True** if it would be valid to create a link between the specified source and target nodes. This checks that both objects belong to this diagram, that the source can supply a link and the target can receive a link, and that creating such a link will not cause a circularity in the diagram.

Exceptions:

OwnerException : if the source or target node are not owned by this diagram

d.iterator() : Iterator

Returns an iterator over the **Node** objects contained in this diagram. The behaviour of the iterator if the diagram is modified between calls of **next()** is undefined.

Note: The iterator returns "top-level" nodes and does not recursively descend into any composite nodes.

d.link(source, target)

source (Node) : the source node

target (Node) : the target node

Creates a new link between the source and the target.

Exceptions:

OwnerException : if any node parameters are not owned by this diagram

ObjectLockedException : if the diagram is currently locked

InvalidEditException : if the connection would be invalid

d.link(source, targets)

source (Node) : the source node

targets (List) : the list of target nodes

Creates new links between the source and each target node in the supplied list.

Exceptions:

OwnerException : if any node parameters are not owned by this diagram

ObjectLockedException : if the diagram is currently locked

InvalidEditException : if a connection would be invalid

ClassCastException : if targets does not contain instances of **Node**

d.linkBetween(inserted, source, target)

inserted (Node) : the node to be inserted

source (Node) : the source node

target (Node) : the target node

Connects a **Node** between two other instances and sets the position of the inserted node to be between those. Any direct link between the source and target is removed first. If any link would be invalid (the source is a terminal node, the target is a source node or the target cannot accept any more links), a **ModelerException** is thrown and no changes are made to the diagram.

Exceptions:

OwnerException : if any of the nodes are not owned by the diagram

ObjectLockedException : if the diagram has been locked because of another operation

InvalidEditException : if the edit is invalid e.g., the target already has the maximum number of input connections

d.linkPath(path)

path (List) : the set of **Node** instances

Creates a new path between **Node** instances. The first node is linked to the second, the second is linked to the third etc. If any link would be invalid (for example, the nodes are already linked, the source is a terminal node, the target is a source node or the target cannot accept any more links), a **ModelerException** is thrown and no changes are made to the diagram.

Exceptions:

OwnerException : if any objects in the path are not owned by this diagram

ObjectLockedException : if the diagram is currently locked

InvalidEditException : if a link between two adjacent nodes in the path cannot be created.

d.linkUpdater(updater, updatable)

updater (Node) : the updater node

updatable (Node) : the updatable node

Creates a new update link between the updater and the updatable. It is expected that updater implements the **Updater** interface while the updatable implements the **Updatable** interface.

Exceptions:

OwnerException : if any node parameters are not owned by this diagram

ObjectLockedException : if the diagram is currently locked

InvalidEditException : if the connection would be invalid

d.predecessorAt(node, index) : Node

node (Node) : the node

index (int) : which predecessor to return

Returns the specified immediate predecessor of the supplied node or **None** if the index is out of bounds.

Exceptions:

OwnerException : if the node is not owned by this diagram

d.predecessorCount(node) : int

node (Node) : the node

Returns the number of immediate predecessors of the supplied node.

Exceptions:

OwnerException : if the node is not owned by this diagram

d.predecessors(node) : List

node (Node) : the node

Returns the immediate predecessors of the supplied node.

Exceptions:

OwnerException : if the node is not owned by this diagram

d.replace(originalNode, replacementNode, discardOriginal)

originalNode (Node) : the node to be replaced

replacementNode (Node) : the new node

discardOriginal (boolean) : if **True**, the id of the original node is assigned to the new node and the original node is automatically deleted from the diagram. If **False**, the original node is retained and the replacement node id is unchanged

Replaces the specified node from this diagram. The both nodes must be owned by this diagram.

Exceptions:

OwnerException : if any node is not owned by this diagram
ObjectLockedException : if the diagram is currently locked
InvalidEditException : is the link operation is invalid
d.setEnabled(node, enabled)

node (Node) : the node

enabled (boolean) : whether the node should be enabled

Sets the enabled state of the supplied node.

Exceptions:

OwnerException : if the node is not owned by this diagram

d.size() : int

Returns the number of **Node** objects contained in this diagram.

d.successorAt(node, index) : Node

node (Node) : the successor

index (int) : which successor to return

Returns the specified immediate successor of the supplied node or **None** if the index is out of bounds.

Exceptions:

OwnerException : if the node is not owned by this diagram

d.successorCount(node) : int

node (Node) : the node

Returns the number of immediate successors of the supplied node.

Exceptions:

OwnerException : if the node is not owned by this diagram

d.successors(node) : List

node (Node) : the node

Returns the immediate successors of the supplied node.

Exceptions:

OwnerException : if the node is not owned by this diagram

d.unlink(source, target)

source (Node) : the source node

target (Node) : the target node

Removes any direct link between the source and the target.

Exceptions:

OwnerException : if any node parameters are not owned by this diagram

ObjectLockedException : if the diagram is currently locked

d.unlink(source, targets)

source (Node) : the source node

targets (List) : the list of target nodes

Removes any direct links between the source and each object in the targets list.

Exceptions:

OwnerException : if any node parameters are not owned by this diagram

ObjectLockedException : if the diagram is currently locked

ClassCastException : if targets does not contain instances of **Node**

d.unlinkPath(path)

path (List) : the list of **Node** instances

Removes any path that exists between **Node** instances. If no link exists between two adjacent nodes in the path, these are silently ignored i.e., no exception will be thrown.

Exceptions:

ObjectLockedException : if the diagram is currently locked

OwnerException : if any objects in the path are not owned by this diagram

d.unlinkUpdater(updater, updatable)

updater (Node) : the updater node

updatable (Node) : the target node

Removes any update link between the updater and the updatable. It is expected that updater implements the **Updater** interface while the updatable implements the **Updatable** interface.

Exceptions:

OwnerException : if any node parameters are not owned by this diagram

ObjectLockedException : if the diagram is currently locked

Related information

- [Streams and SuperNodes](#)

NodeFilter Objects

This is used to define arbitrary criteria for filtering nodes e.g. during searches.

n.accept(node) : boolean

node (Node) : the node

Returns **True** if the node should be included by the filter.

Related information

- [Streams and SuperNodes](#)

Stream Objects

Subclass of **Diagram**, **PropertiedObject**, **ParameterProvider**.

This is the top-level container used to assemble **Node** objects into a connected "flow". It also provides the environment for setting which may be used to modify node behaviour.

s.close()

Closes the current stream. If the stream is already closed, this method does nothing. No further operations can be applied to a closed stream.

s.getContentProvider() : ContentProvider

Returns the **ContentProvider** for this stream. The content provider manages additional content on behalf of applications.

```
s.getGlobalValues() : GlobalValues
```

Returns the global values computed for this stream. Global values are constructed and updated by executing a Set Globals node.

```
s.getID() : string
```

Returns the temporary session ID of this object. A new ID is allocated each time a new stream is created or opened and the ID is not persisted when the stream is saved. This means that if the same persisted object is re-opened multiple times, each object will have a different ID.

```
s.getServerConnectionDescriptor() : ServerConnectionDescriptor
```

Returns the **ServerConnectionDescriptor** used to connect this stream to a server or **None** if the stream has not yet been connected or if the owner session was not created using **SessionFactory**.

```
s.isClosed() : boolean
```

Returns **True** if the stream has been closed, **False** otherwise.

```
s.isConnected() : boolean
```

Returns **True** if this has a server connection.

```
s.isExportable(format) : boolean  
format (FileFormat) :
```

Returns **True** if this stream can be exported using the supplied **FileFormat** or **False** otherwise.

```
s.runAll(results) : ExecutionHandle
```

results (Collection) : an empty collection that will contain any built objects once execution has completed

Executes the stream synchronously and waits for it to complete. Returns an **ExecutionHandle** which can be used to access the exit status and any result from the task.

Exceptions:

OwnerException : if the stream was not created by this session

ObjectLockedException : if the stream is locked

ServerConnectionException : if the stream is not connected to a server

SessionException : if the session is already running another task, cannot execute the task or if execution completes in a state other than **SUCCESS**

```
s.runScript(results) : ExecutionHandle
```

results (Collection) : an empty collection that will contain any built objects once execution has completed

Executes the stream script synchronously and waits for it to complete. The stream script is always run regardless of whether script execution is set as the default behaviour. Returns an **ExecutionHandle** which can be used to access the exit status and any result from the task.

Exceptions:

OwnerException : if the stream was not created by this session

ObjectLockedException : if the stream is locked

ServerConnectionException : if the stream is not connected to a server

SessionException : if the session is already running another task, cannot execute the task or if execution completes in a state other than **SUCCESS**

```
s.runSelected(nodes, results) : ExecutionHandle
```

nodes (Node[]) : the array of **Node** objects to be executed

results (Collection) : an empty collection that will contain any built objects once execution has completed

Executes the supplied array of nodes synchronously and waits for them to complete. There must be at least one node in the array. Returns an **ExecutionHandle** which can be used to access the exit status and any result from the task.

Exceptions:

OwnerException : if the nodes are not all owned by this stream

ObjectLockedException : if the stream is locked

ServerConnectionException : if the stream is not connected to a server

SessionException : if the session is already running a stream, another task, cannot execute the task or if execution completes in a state other than **SUCCESS**

IllegalArgumentException : if the array is empty

Related information

- [Streams and SuperNodes](#)

PublishedImage Objects

The result of publishing a **DataWriter**.

p.getImageContent() : byte[]

Returns the contents of the image file.

p.getMetadataContent() : byte[]

Returns the contents of the metadata file, or **None** if there is no metadata file.

p.getOutputDataModel() : DataModel

Returns a data model which describes the fields exported by the image. This is the server-side data model and differs from the export data model of the published node in that fields typically have known storage but default measure.

p.getParameterContent() : byte[]

Returns the contents of the parameter file.

Related information

- [Streams and SuperNodes](#)

ReportBuilder Objects

Subclass of **DocumentBuilder**.

This encapsulates the functionality of a node that builds a report.

r.getReportOutput() : ReportOutput

Returns the most recent **ReportOutput** produced by this node, or **None** if this **ReportBuilder** has either not been executed before or failed to produce an object during the most recent execution.

Related information

- [Streams and SuperNodes](#)

ReportOutput Objects

Subclass of **DocumentOutput**.

This encapsulates the concrete results of executing **DocumentBuilder** nodes that produce report-type outputs such as quality reports.

Related information

- [Streams and SuperNodes](#)

RowSetBuilder Objects

Subclass of **DocumentBuilder**.

This encapsulates the functionality of a node that builds an output based on a **RowSet**.

r.getRowSetOutput() : RowSetOutput

Returns the most recent **RowSetOutput** produced by this node, or **None** if this **RowSetBuilder** has either not been executed before or failed to produce an object during the most recent execution.

Related information

- [Streams and SuperNodes](#)

RowSetOutput Objects

Subclass of **DocumentOutput**.

This encapsulates the concrete results of executing **DocumentBuilder** nodes that produce objects that are based on underlying **RowSet** such as tables and matrices.

r.getRowSet() : RowSet

Returns the **RowSet** underlying this output object.

Related information

- [Streams and SuperNodes](#)

TerminalNode Objects

Subclass of **Node**.

This encapsulates the functionality of any node that terminates a particular sequence of nodes.

Related information

- [Streams and SuperNodes](#)

Updatable Objects

Subclass of **Node**.

This represents **Node** objects that can be updated by **Updating** objects. Updating can happen either by modifying the contents of an updatable object, or by replacing it in the stream. An updatable object can only be associated with one updater at a time.

u.getUpdater() : Node

Returns the **Node** that is updating this one or **None** if there is no updater.

u.getUpdaterID() : string

Returns the id of the node that is updating this one or the empty string if there is no updater.

u.isUpdatingEnabled() : boolean

Returns **True** if this node can be updated by an updater. This value can be set or returned regardless of whether an updater has been specified.

u.setUpdatingEnabled(canUpdate)
canUpdate (boolean) :

Sets whether this node can be updated by any associated updater.

Related information

- [Streams and SuperNodes](#)
-

Updater Objects

Subclass of **Node**.

This represents **Node** objects that update **Updatable** objects. Updating can happen either by modifying the contents of an updatable object, or by replacing it in the stream. An updater can have multiple updatable objects associated with it.

u.getUpdatableCount() : int

Returns the number of **Updatable** nodes that this is updating.

u.getUpdatables() : Iterator

Returns an iterator for the **Updatable** nodes that this is updating.

Related information

- [Streams and SuperNodes](#)
-

In-Database Mining

- [Database Modeling Overview](#)
-

Database Modeling Overview

IBM® SPSS® Modeler Server supports integration with data mining and modeling tools that are available from database vendors, including IBM Netezza, Oracle Data Miner, and Microsoft Analysis Services. You can build, score, and store models inside the database—all from within the IBM SPSS Modeler application. This allows you to combine the analytical capabilities and ease-of-use of IBM SPSS Modeler with the power and performance of a database, while taking advantage of database-native algorithms provided by these vendors. Models are built inside the database, which can then be browsed and scored through the IBM SPSS Modeler interface in the normal manner and can be deployed using IBM SPSS Modeler Solution Publisher if needed. Supported algorithms are on the Database Modeling palette in IBM SPSS Modeler.

Using IBM SPSS Modeler to access database-native algorithms offers several advantages:

- In-database algorithms are often closely integrated with the database server and may offer improved performance.
- Models built and stored “in database” may be more easily deployed to and shared with any application that can access the database.

SQL generation. In-database modeling is distinct from SQL generation, otherwise known as “SQL pushback”. This feature allows you to generate SQL statements for native IBM SPSS Modeler operations that can be “pushed back” to (that is, executed in) the database in order to improve performance. For example, the Merge, Aggregate, and Select nodes all generate SQL code that can be pushed back to the database in this manner. Using SQL generation in combination with database modeling may result in streams that can be run from start to finish in the database, resulting in significant performance gains over streams run in IBM SPSS Modeler.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > About > Additional Details

If connectivity is enabled, you see the option Server Enablement in the License Status tab.

See [Connecting to IBM SPSS Modeler Server](#) for more information.

For information on supported algorithms, see the subsequent sections on specific vendors.

- [What you need](#)
- [Model Building](#)
- [Data Preparation](#)
- [Model scoring](#)
- [Exporting and saving database models](#)

- [Model Consistency](#)
 - [Viewing and Exporting Generated SQL](#)
-

What you need

To perform database modeling, you need the following setup:

- An ODBC connection to an appropriate database, with required analytical components installed (Microsoft Analysis Services or Oracle Data Miner).
- In IBM® SPSS® Modeler, database modeling must be enabled in the Helper Applications dialog box (Tools > Helper Applications).
- The Generate SQL and SQL Optimization settings should be enabled in the User Options dialog box in IBM SPSS Modeler as well as on IBM SPSS Modeler Server (if used). Note that SQL optimization is not strictly required for database modeling to work but is highly recommended for performance reasons.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > About > Additional Details

If connectivity is enabled, you see the option Server Enablement in the License Status tab.

See [Connecting to IBM SPSS Modeler Server](#) for more information.

For detailed information, see the subsequent sections on specific vendors.

Model Building

The process of building and scoring models by using database algorithms is similar to other types of data mining in IBM® SPSS® Modeler. The general process of working with nodes and modeling "nuggets" is similar to any other stream when working in IBM SPSS Modeler. The only difference is that the actual processing and model building is pushed back to the database.

A Database modelling stream is conceptually identical to other data streams in IBM SPSS Modeler; however, this stream performs all operations in a database, including, for example, model-building using the Microsoft Decision Tree node. When you run the stream, IBM SPSS Modeler instructs the database to build and store the resulting model, and details are downloaded to IBM SPSS Modeler. In-database execution is indicated by the use of purple-shaded nodes in the stream.

Data Preparation

Whether or not database-native algorithms are used, data preparations should be pushed back to the database whenever possible in order to improve performance.

- If the original data is stored in the database, the goal is to keep it there by making sure that all required upstream operations can be converted to SQL. This will prevent data from being downloaded to IBM® SPSS® Modeler—avoiding a bottleneck that might nullify any gains—and allow the entire stream to be run in the database.
 - If the original data are *not* stored in the database, database modeling can still be used. In this case, the data preparation is conducted in IBM SPSS Modeler and the prepared dataset is automatically uploaded to the database for model building.
-

Model scoring

Models generated from IBM® SPSS® Modeler by using in-database mining are different from regular IBM SPSS Modeler models. Although they appear in the Model manager as generated model "nuggets," they are actually remote models held on the remote data mining or database server. What you see in IBM SPSS Modeler are simply references to these remote models. In other words, the IBM SPSS Modeler model you see is a "hollow" model that contains information such as the database server hostname, database name, and the model name. This is an important distinction to understand as you browse and score models created using database-native algorithms.

Once you have created a model, you can add it to the stream for scoring like any other generated model in IBM SPSS Modeler. All scoring is done within the database, even if upstream operations are not. (Upstream operations may still be pushed back to the database if possible to improve performance, but this is not a requirement for scoring to take place.) You can also browse the generated model in most cases using the standard browser provided by the database vendor.

For both browsing and scoring, a live connection to the server running Oracle Data Miner or Microsoft Analysis Services is required.

Viewing results and specifying settings

To view results and specify settings for scoring, double-click the model on the stream canvas. Alternatively, you can right-click the model and choose Browse or Edit. Specific settings depend on the type of model.

Exporting and saving database models

Database models and summaries can be exported from the model browser in the same manner as other models created in IBM® SPSS® Modeler, using options on the File menu.

1. From the File menu in the model browser, choose any of the following options:

- Export Text exports the model summary to a text file
- Export HTML exports the model summary to an HTML file
- Export PMML (supported for IBM Db2 IM models only) exports the model as predictive model markup language (PMML), which can be used with other PMML-compatible software.

Note: You can also save a generated model by choosing Save Node from the File menu.

Model Consistency

For each generated database model, IBM® SPSS® Modeler stores a description of the model structure, along with a reference to the model with the same name that is stored within the database. The Server tab of a generated model displays a unique key generated for that model, which matches the actual model in the database.

IBM SPSS Modeler uses this randomly generated key to check that the model is still consistent. This key is stored in the description of a model when it is built. It is a good idea to check that keys match before running a deployment stream.

1. To check the consistency of the model stored in the database by comparing its description with the random key stored by IBM SPSS Modeler, click the Check button. If the database model cannot be found or the key does not match, an error is reported.

Related information

- [Analysis Services Mining Examples](#)

Viewing and Exporting Generated SQL

The generated SQL code can be previewed prior to execution, which may be useful for debugging purposes.

Database Modeling with Microsoft Analysis Services

- [IBM SPSS Modeler and Microsoft Analysis Services](#)
- [Building Models with Analysis Services](#)
- [Scoring Analysis Services Models](#)
- [Analysis Services Mining Examples](#)

IBM SPSS Modeler and Microsoft Analysis Services

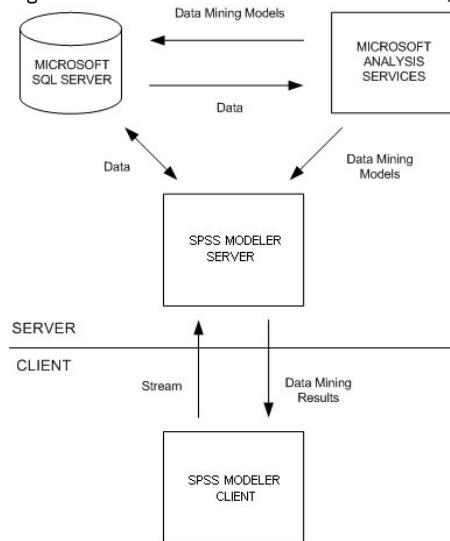
IBM® SPSS® Modeler supports integration with Microsoft SQL Server Analysis Services. This functionality is implemented as modeling nodes in IBM SPSS Modeler and is available from the Database Modeling palette. If the palette is not visible, you can activate it by enabling MS Analysis Services integration, available on the Microsoft tab, from the Helper Applications dialog box. See the topic [Enabling Integration with Analysis Services](#) for more information.

IBM SPSS Modeler supports integration of the following Analysis Services algorithms:

- Decision Trees
- Clustering
- Association Rules
- Naïve Bayes
- Linear Regression
- Neural Network
- Logistic Regression
- Time Series
- Sequence Clustering

The following diagram illustrates the flow of data from client to server where in-database mining is managed by IBM SPSS Modeler Server. Model building is performed using Analysis Services. The resulting model is stored by Analysis Services. A reference to this model is maintained within IBM SPSS Modeler streams. The model is then downloaded from Analysis Services to either Microsoft SQL Server or IBM SPSS Modeler for scoring.

Figure 1. Data flow between IBM SPSS Modeler, Microsoft SQL Server, and Microsoft Analysis Services during model building



Note: The IBM SPSS Modeler Server is not required, though it can be used. IBM SPSS Modeler client is capable of processing in-database mining calculations itself.

- [Requirements for Integration with Microsoft Analysis Services](#)
- [Enabling Integration with Analysis Services](#)

Related information

- [Requirements for Integration with Microsoft Analysis Services](#)
- [Enabling Integration with Analysis Services](#)
- [Building Models with Analysis Services](#)
- [Scoring Analysis Services Models](#)

Requirements for Integration with Microsoft Analysis Services

The following are prerequisites for conducting in-database modeling using Analysis Services algorithms with IBM® SPSS® Modeler. You may need to consult with your database administrator to ensure that these conditions are met.

- IBM SPSS Modeler running against an IBM SPSS Modeler Server installation (distributed mode) on Windows. UNIX platforms are not supported in this integration with Analysis Services.
Important: IBM SPSS Modeler users must configure an ODBC connection using the SQL Native Client driver available from Microsoft at the URL listed below under *Additional IBM SPSS Modeler Server Requirements*. *The driver provided with the IBM SPSS Data Access Pack (and typically recommended for other uses with IBM SPSS Modeler) is not recommended for this purpose.* The driver should be configured to use SQL Server With Integrated Windows Authentication enabled, since IBM SPSS Modeler does not support SQL Server authentication. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.
- SQL Server must be installed, although not necessarily on the same host as IBM SPSS Modeler. IBM SPSS Modeler users must have sufficient permissions to read and write data and drop and create tables and views.

Note: SQL Server Enterprise Edition is recommended. The Enterprise Edition provides additional flexibility by providing advanced parameters to tune algorithm results. The Standard Edition version provides the same parameters but does not allow users to edit some of the advanced parameters.

- Microsoft SQL Server Analysis Services must be installed on the same host as SQL Server.

Additional IBM SPSS Modeler Server Requirements

To use Analysis Services algorithms with IBM SPSS Modeler Server, the following components must be installed on the IBM SPSS Modeler Server host machine.

Note: If SQL Server is installed on the same host as IBM SPSS Modeler Server, these components will already be available.

- Microsoft SQL Server Analysis Services 10.0 OLE DB Provider (be sure to select the correct variant for your operating system)
- Microsoft SQL Server Native Client (be sure to select the correct variant for your operating system)
- If you are using Microsoft SQL Server 2008 or 2012, you might also require Microsoft Core XML Services (MSXML) 6.0.

To download these components, go to www.microsoft.com/downloads, search for .NET Framework or (for all other components) SQL Server Feature Pack, and select the latest pack for your version of SQL Server.

These may require other packages to be installed first, which should also be available on the Microsoft Downloads web site.

Additional IBM SPSS Modeler Requirements

To use Analysis Services algorithms with IBM SPSS Modeler, the same components must be installed as above, with the addition of the following at the client:

- Microsoft SQL Server Datamining Viewer Controls (be sure to select the correct variant for your operating system) - this also requires:
- Microsoft ADOMD.NET

To download these components, go to www.microsoft.com/downloads, search for SQL Server Feature Pack, and select the latest pack for your version of SQL Server.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > About > Additional Details

If connectivity is enabled, you see the option Server Enablement in the License Status tab.

See [Connecting to IBM SPSS Modeler Server](#) for more information.

Related information

- [Enabling Integration with Analysis Services](#)
 - [Building Models with Analysis Services](#)
 - [Scoring Analysis Services Models](#)
-

Enabling Integration with Analysis Services

To enable IBM® SPSS® Modeler integration with Analysis Services, you will need to configure SQL Server and Analysis Services, create an ODBC source, enable the integration in the IBM SPSS Modeler Helper Applications dialog box, and enable SQL generation and optimization.

Note: Microsoft SQL Server and Microsoft Analysis Services must be available. See the topic [Requirements for Integration with Microsoft Analysis Services](#) for more information.

Configuring SQL Server

Configure SQL Server to allow scoring to take place within the database.

1. Create the following registry key on the SQL Server host machine:

`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\MSOLAP`

2. Add the following **DWORD** value to this key:

`AllowInProcess 1`

3. Restart SQL Server after making this change.

Configuring Analysis Services

Before IBM SPSS Modeler can communicate with Analysis Services, you must first manually configure two settings in the Analysis Server Properties dialog box:

1. Log in to the Analysis Server through the MS SQL Server Management Studio.
2. Access the Properties dialog box by right-clicking the server name and choosing Properties.
3. Select the Show Advanced (All) Properties check box.
4. Change the following properties:
 - Change the value for **DataMining\AllowAdHocOpenRowsetQueries** to **True** (the default value is **False**).
 - Change the value for **DataMining\AllowProvidersInOpenRowset** to **[all]** (there is no default value).

Creating an ODBC DSN for SQL Server

To read or write to a database, you must have an ODBC data source installed and configured for the relevant database, with read or write permissions as needed. The Microsoft SQL Native Client ODBC driver is required and automatically installed with SQL Server. *The driver provided with the IBM SPSS Data Access Pack (and typically recommended for other uses with IBM SPSS Modeler) is not recommended for this purpose.* If IBM SPSS Modeler and SQL Server reside on different hosts, you can download the Microsoft SQL Native Client ODBC driver. See the topic [Requirements for Integration with Microsoft Analysis Services](#) for more information.

If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

1. Using the Microsoft SQL Native Client ODBC driver, create an ODBC DSN that points to the SQL Server database used in the data mining process. The remaining default driver settings should be used.
2. For this DSN, ensure that With Integrated Windows Authentication is selected.
 - If IBM SPSS Modeler and IBM SPSS Modeler Server are running on different hosts, create the same ODBC DSN on each of the hosts. Ensure that the same DSN name is used on each host.

Enabling the Analysis Services Integration in IBM SPSS Modeler

To enable IBM SPSS Modeler to use Analysis Services, you must first provide server specifications in the Helper Applications dialog box.

1. From the IBM SPSS Modeler menus choose:
Tools > Options > Helper Applications
2. Click the Microsoft tab.
 - **Enable Microsoft Analysis Services Integration.** Enables the Database Modeling palette (if not already displayed) at the bottom of the IBM SPSS Modeler window and adds the nodes for Analysis Services algorithms.
 - **Analysis Server Host.** Specify the name of the machine on which Analysis Services is running.
 - **Analysis Server Database.** Select the desired database by clicking the ellipsis (...) button to open a subdialog box in which you can choose from available databases. The list is populated with databases available to the specified Analysis server. Since Microsoft Analysis Services stores data mining models within named databases, you should select the appropriate database in which Microsoft models built by IBM SPSS Modeler are stored.
 - **SQL Server Connection.** Specify the DSN information used by the SQL Server database to store the data that are passed into the Analysis server. Choose the ODBC data source that will be used to provide the data for building Analysis Services data mining models. If you are building Analysis Services models from data supplied in flat files or ODBC data sources, the data will be automatically uploaded to a temporary table created in the SQL Server database to which this ODBC data source points.
 - **Warn when about to overwrite a data mining model.** Select to ensure that models stored in the database are not overwritten by IBM SPSS Modeler without warning.

Note: Settings made in the Helper Applications dialog box can be overridden inside the various Analysis Services nodes.

Enabling SQL Generation and Optimization

1. From the IBM SPSS Modeler menus choose:
Tools > Stream Properties > Options
2. Click the Optimization option in the navigation pane.
3. Confirm that the Generate SQL option is enabled. This setting is required for database modeling to function.
4. Select Optimize SQL Generation and Optimize other execution (not strictly required but strongly recommended for optimized performance).

Related information

- [IBM SPSS Modeler and Microsoft Analysis Services](#)
- [Requirements for Integration with Microsoft Analysis Services](#)
- [Building Models with Analysis Services](#)
- [Scoring Analysis Services Models](#)

Building Models with Analysis Services

Analysis Services model building requires that the training dataset be located in a table or view within the SQL Server database. If the data is not located in SQL Server or need to be processed in IBM® SPSS® Modeler as part of data preparation that cannot be performed in SQL Server, the data is automatically uploaded to a temporary table in SQL Server before model building.

- [Managing Analysis Services Models](#)
- [Settings Common to All Algorithm Nodes](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naïve Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Time Series Node](#)
- [MS Sequence Clustering Node](#)

Related information

- [IBM SPSS Modeler and Microsoft Analysis Services](#)
- [Requirements for Integration with Microsoft Analysis Services](#)
- [Enabling Integration with Analysis Services](#)
- [Scoring Analysis Services Models](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Model Options](#)
- [MS Time Series Expert Options](#)
- [MS Time Series Settings Options](#)
- [MS Sequence Clustering Fields Options](#)
- [MS Sequence Clustering Expert Options](#)
- [Settings Common to All Analysis Services Models](#)
- [MS Time Series Model Nugget](#)
- [MS Time Series Model Nugget Server Tab](#)
- [MS Time Series Model Nugget Settings Tab](#)
- [MS Sequence Clustering Model Nugget](#)
- [Exporting Models and Generating Nodes](#)

Managing Analysis Services Models

Building an Analysis Services model via IBM® SPSS® Modeler creates a model in IBM SPSS Modeler and creates or replaces a model in the SQL Server database. The IBM SPSS Modeler model references the content of a database model stored on a database server. IBM SPSS Modeler can perform consistency checking by storing an identical generated model key string in both the IBM SPSS Modeler model and the SQL Server model.

	The MS Decision Tree modeling node is used in predictive modeling of both categorical and continuous attributes. For categorical attributes, the node makes predictions based on the relationships between input columns in a dataset. For example, in a scenario to predict which customers are likely to purchase a bicycle, if nine out of ten younger customers buy a bicycle but only two out of ten older customers do so, the node infers that age is a good predictor of bicycle purchase. The decision tree makes predictions based on this tendency toward a particular outcome. For continuous attributes, the algorithm uses linear regression to determine where a decision tree splits. If more than one column is set to predictable, or if the input data contains a nested table that is set to predictable, the node builds a separate decision tree for each predictable column.
	The MS Clustering modeling node uses iterative techniques to group cases in a dataset into clusters that contain similar characteristics. These groupings are useful for exploring data, identifying anomalies in the data, and creating predictions. Clustering models identify relationships in a dataset that you might not logically derive through casual observation. For example, you can logically discern that people who commute to their jobs by bicycle do not typically live a long distance from where they work. The algorithm, however, can find other characteristics about bicycle commuters that are not as obvious. The clustering node differs from other data mining nodes in that no target field is specified. The clustering node trains the model strictly from the relationships that exist in the data and from the clusters that the node identifies.
	The MS Association Rules modeling node is useful for recommendation engines. A recommendation engine recommends products to customers based on items they have already purchased or in which they have indicated an interest. Association models are built on datasets that contain identifiers both for individual cases and for the items that the cases contain. A group of items in a case is called an itemset . An association model is made up of a series of itemsets and the rules that

	<p>describe how those items are grouped together within the cases. The rules that the algorithm identifies can be used to predict a customer's likely future purchases, based on the items that already exist in the customer's shopping cart.</p>
	<p>The MS Naive Bayes modeling node calculates the conditional probability between target and predictor fields and assumes that the columns are independent. The model is termed naïve because it treats all proposed prediction variables as being independent of one another. This method is computationally less intense than other Analysis Services algorithms and therefore useful for quickly discovering relationships during the preliminary stages of modeling. You can use this node to do initial explorations of data and then apply the results to create additional models with other nodes that may take longer to compute but give more accurate results.</p>
	<p>The MS Linear Regression modeling node is a variation of the Decision Trees node, where the MINIMUM_LEAF_CASES parameter is set to be greater than or equal to the total number of cases in the dataset that the node uses to train the mining model. With the parameter set in this way, the node will never create a split and therefore performs a linear regression.</p>
	<p>The MS Neural Network modeling node is similar to the MS Decision Tree node in that the MS Neural Network node calculates probabilities for each possible state of the input attribute when given each state of the predictable attribute. You can later use these probabilities to predict an outcome of the predicted attribute, based on the input attributes.</p>
	<p>The MS Logistic Regression modeling node is a variation of the MS Neural Network node, where the HIDDEN_NODE_RATIO parameter is set to 0. This setting creates a neural network model that does not contain a hidden layer and therefore is equivalent to logistic regression.</p>
	<p>The MS Time Series modeling node provides regression algorithms that are optimized for the forecasting of continuous values, such as product sales, over time. Whereas other Microsoft algorithms, such as decision trees, require additional columns of new information as input to predict a trend, a time series model does not. A time series model can predict trends based only on the original dataset that is used to create the model. You can also add new data to the model when you make a prediction and automatically incorporate the new data in the trend analysis. See the topic MS Time Series Node for more information.</p>
	<p>The MS Sequence Clustering modeling node identifies ordered sequences in data, and combines the results of this analysis with clustering techniques to generate clusters based on the sequences and other attributes. See the topic MS Sequence Clustering Node for more information.</p>

You can access each node from the Database Modeling palette at the bottom of the IBM SPSS Modeler window.

Related information

- [Building Models with Analysis Services](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Node](#)
- [MS Time Series Model Options](#)
- [MS Time Series Expert Options](#)
- [MS Time Series Settings Options](#)
- [MS Sequence Clustering Node](#)
- [MS Sequence Clustering Fields Options](#)
- [MS Sequence Clustering Expert Options](#)
- [Settings Common to All Analysis Services Models](#)
- [MS Time Series Model Nugget](#)
- [MS Time Series Model Nugget Server Tab](#)
- [MS Time Series Model Nugget Settings Tab](#)
- [MS Sequence Clustering Model Nugget](#)
- [Exporting Models and Generating Nodes](#)

Settings Common to All Algorithm Nodes

The following settings are common to all Analysis Services algorithms.

- [Server Options](#)
- [Model Options](#)

Related information

- [Server Options](#)

- [Model Options](#)
-

Server Options

On the Server tab, you can configure the Analysis server host, database, and the SQL Server data source. Options specified here overwrite those specified on the Microsoft tab in the Helper Applications dialog box. See the topic [Enabling Integration with Analysis Services](#) for more information.

Note: A variation of this tab is also available when scoring Analysis Services models. See the topic [Analysis Services Model Nugget Server Tab](#) for more information.

Related information

- [Model Options](#)
-

Model Options

In order to build the most basic model, you need to specify options on the Model tab before proceeding. Scoring method and other advanced options are available on the Expert tab.

The following basic modeling options are available:

Model name. Specifies the name assigned to the model that is created when the node is executed.

- **Auto.** Generates the model name automatically based on the target or ID field names or the name of the model type in cases where no target is specified (such as clustering models).
- **Custom.** Allows you to specify a custom name for the model created.

Use partitioned data. Splits the data into separate subsets or samples for training, testing, and validation based on the current partition field. Using one sample to create the model and a separate sample to test it may provide an indication of how well the model will generalize to larger datasets that are similar to the current data. If no partition field is specified in the stream, this option is ignored.

With Drillthrough. If shown, this option enables you to query the model to learn details about the cases included in the model.

Unique field. From the drop-down list, select a field that uniquely identifies each case. Typically, this is an ID field, such as CustomerID.

Related information

- [Server Options](#)
-

MS Decision Tree Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

Related information

- [Managing Analysis Services Models](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Node](#)
- [MS Time Series Model Options](#)
- [MS Time Series Expert Options](#)
- [MS Time Series Settings Options](#)
- [MS Sequence Clustering Node](#)

- [MS Sequence Clustering Fields Options](#)
 - [MS Sequence Clustering Expert Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Clustering Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

Related information

- [Managing Analysis Services Models](#)
 - [MS Decision Tree Expert Options](#)
 - [MS Naive Bayes Expert Options](#)
 - [MS Linear Regression Expert Options](#)
 - [MS Neural Network Expert Options](#)
 - [MS Logistic Regression Expert Options](#)
 - [MS Association Rules Node](#)
 - [MS Association Rules Expert Options](#)
 - [MS Time Series Node](#)
 - [MS Time Series Model Options](#)
 - [MS Time Series Expert Options](#)
 - [MS Time Series Settings Options](#)
 - [MS Sequence Clustering Node](#)
 - [MS Sequence Clustering Fields Options](#)
 - [MS Sequence Clustering Expert Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Naive Bayes Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

Related information

- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Node](#)
- [MS Time Series Model Options](#)
- [MS Time Series Expert Options](#)
- [MS Time Series Settings Options](#)
- [MS Sequence Clustering Node](#)
- [MS Sequence Clustering Fields Options](#)
- [MS Sequence Clustering Expert Options](#)
- [Settings Common to All Analysis Services Models](#)
- [MS Time Series Model Nugget](#)

- [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Linear Regression Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

Related information

- [Managing Analysis Services Models](#)
 - [MS Decision Tree Expert Options](#)
 - [MS Clustering Expert Options](#)
 - [MS Naive Bayes Expert Options](#)
 - [MS Neural Network Expert Options](#)
 - [MS Logistic Regression Expert Options](#)
 - [MS Association Rules Node](#)
 - [MS Association Rules Expert Options](#)
 - [MS Time Series Node](#)
 - [MS Time Series Model Options](#)
 - [MS Time Series Expert Options](#)
 - [MS Time Series Settings Options](#)
 - [MS Sequence Clustering Node](#)
 - [MS Sequence Clustering Fields Options](#)
 - [MS Sequence Clustering Expert Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Neural Network Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

Related information

- [Managing Analysis Services Models](#)
 - [MS Decision Tree Expert Options](#)
 - [MS Clustering Expert Options](#)
 - [MS Naive Bayes Expert Options](#)
 - [MS Linear Regression Expert Options](#)
 - [MS Logistic Regression Expert Options](#)
 - [MS Association Rules Node](#)
 - [MS Association Rules Expert Options](#)
 - [MS Time Series Node](#)
 - [MS Time Series Model Options](#)
 - [MS Time Series Expert Options](#)
 - [MS Time Series Settings Options](#)
 - [MS Sequence Clustering Node](#)
 - [MS Sequence Clustering Fields Options](#)
 - [MS Sequence Clustering Expert Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Logistic Regression Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

Related information

- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Node](#)
- [MS Time Series Model Options](#)
- [MS Time Series Expert Options](#)
- [MS Time Series Settings Options](#)
- [MS Sequence Clustering Node](#)
- [MS Sequence Clustering Fields Options](#)
- [MS Sequence Clustering Expert Options](#)
- [Settings Common to All Analysis Services Models](#)
- [MS Time Series Model Nugget](#)
- [MS Time Series Model Nugget Server Tab](#)
- [MS Time Series Model Nugget Settings Tab](#)
- [MS Sequence Clustering Model Nugget](#)
- [Exporting Models and Generating Nodes](#)

MS Association Rules Node

The MS Association Rules modeling node is useful for recommendation engines. A recommendation engine recommends products to customers based on items they have already purchased or in which they have indicated an interest. Association models are built on datasets that contain identifiers both for individual cases and for the items that the cases contain. A group of items in a case is called an **itemset**.

An association model is made up of a series of itemsets and the rules that describe how those items are grouped together within the cases. The rules that the algorithm identifies can be used to predict a customer's likely future purchases, based on the items that already exist in the customer's shopping cart.

For tabular format data, the algorithm creates scores that represent probability (*\$MP-field*) for each generated recommendation (*\$M-field*). For transactional format data, scores are created for support (*\$MS-field*), probability (*\$MP-field*) and adjusted probability (*\$MAP-field*) for each generated recommendation (*\$M-field*).

Requirements

The requirements for a transactional association model are as follows:

- **Unique field.** An association rules model requires a key that uniquely identifies records.
- **ID field.** When building an MS Association Rules model with transactional format data, an ID field that identifies each transaction is required. ID fields can be set to the same as the unique field.
- **At least one input field.** The Association Rules algorithm requires at least one input field.
- **Target field.** When building an MS Association model with transactional data, the target field must be the transaction field, for example products that a user bought.
- [MS Association Rules Expert Options](#)

Related information

- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)

- [MS Logistic Regression Expert Options](#)
 - [MS Time Series Node](#)
 - [MS Time Series Model Options](#)
 - [MS Time Series Expert Options](#)
 - [MS Time Series Settings Options](#)
 - [MS Sequence Clustering Node](#)
 - [MS Sequence Clustering Fields Options](#)
 - [MS Sequence Clustering Expert Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Association Rules Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

Related information

- [Building Models with Analysis Services](#)
 - [Managing Analysis Services Models](#)
 - [MS Decision Tree Expert Options](#)
 - [MS Clustering Expert Options](#)
 - [MS Naive Bayes Expert Options](#)
 - [MS Linear Regression Expert Options](#)
 - [MS Neural Network Expert Options](#)
 - [MS Logistic Regression Expert Options](#)
 - [MS Time Series Node](#)
 - [MS Time Series Model Options](#)
 - [MS Time Series Expert Options](#)
 - [MS Time Series Settings Options](#)
 - [MS Sequence Clustering Node](#)
 - [MS Sequence Clustering Fields Options](#)
 - [MS Sequence Clustering Expert Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Time Series Node

The MS Time Series modeling node supports two types of predictions:

- future
- historical

Future predictions estimate target field values for a specified number of time periods beyond the end of your historical data, and are always performed. **Historical predictions** are estimated target field values for a specified number of time periods for which you have the actual values in your historical data. You can use historical predictions to assess the quality of the model, by comparing the actual historical values with the predicted values. The value of the start point for the predictions determines whether historical predictions are performed.

Unlike the IBM® SPSS® Modeler Time Series node, the MS Time Series node does not need a preceding Time Intervals node. A further difference is that by default, scores are produced only for the predicted rows, not for all the historical rows in the time series data.

Requirements

The requirements for an MS Time Series model are as follows:

- **Single key time field.** Each model must contain one numeric or date field that is used as the case series, defining the time slices that the model will use. The data type for the key time field can be either a datetime data type or a numeric data type. However, the field must contain continuous values, and the values must be unique for each series.
 - **Single target field.** You can specify only one target field in each model. The data type of the target field must have continuous values. For example, you can predict how numeric attributes, such as income, sales, or temperature, change over time. However, you cannot use a field that contains categorical values, such as purchasing status or level of education, as the target field.
 - **At least one input field.** The MS Time Series algorithm requires at least one input field. The data type of the input field must have continuous values. Non-continuous input fields are ignored when building the model.
 - **Dataset must be sorted.** The input dataset must be sorted (on the key time field), otherwise model building will be interrupted with an error.
- [MS Time Series Model Options](#)
 • [MS Time Series Expert Options](#)
 • [MS Time Series Settings Options](#)

Related information

- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Sequence Clustering Node](#)
- [MS Sequence Clustering Fields Options](#)
- [MS Sequence Clustering Expert Options](#)
- [Settings Common to All Analysis Services Models](#)
- [MS Time Series Model Nugget](#)
- [MS Time Series Model Nugget Server Tab](#)
- [MS Time Series Model Nugget Settings Tab](#)
- [MS Sequence Clustering Model Nugget](#)
- [Exporting Models and Generating Nodes](#)

MS Time Series Model Options

Model name. Specifies the name assigned to the model that is created when the node is executed.

- **Auto.** Generates the model name automatically based on the target or ID field names or the name of the model type in cases where no target is specified (such as clustering models).
- **Custom.** Allows you to specify a custom name for the model created.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

With Drillthrough. If shown, this option enables you to query the model to learn details about the cases included in the model.

Unique field. From the drop-down list, select the key time field, which is used to build the time series model.

Related information

- [Building Models with Analysis Services](#)
- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Expert Options](#)
- [MS Time Series Settings Options](#)
- [MS Sequence Clustering Node](#)
- [MS Sequence Clustering Fields Options](#)

- [MS Sequence Clustering Expert Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Time Series Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

If you are making historical predictions, the number of historical steps that can be included in the scoring result is decided by the value of (HISTORIC_MODEL_COUNT * HISTORIC_MODEL_GAP). By default, this limitation is 10, meaning that only 10 historical predictions will be made. In this case, for example, an error occurs if you enter a value of less than -10 for Historical prediction on the Settings tab of the model nugget (see [MS Time Series Model Nugget Settings Tab](#)). If you want to see more historical predictions, you can increase the value of HISTORIC_MODEL_COUNT or HISTORIC_MODEL_GAP, but this will increase the build time for the model.

Related information

- [Building Models with Analysis Services](#)
 - [Managing Analysis Services Models](#)
 - [MS Decision Tree Expert Options](#)
 - [MS Clustering Expert Options](#)
 - [MS Naïve Bayes Expert Options](#)
 - [MS Linear Regression Expert Options](#)
 - [MS Neural Network Expert Options](#)
 - [MS Logistic Regression Expert Options](#)
 - [MS Association Rules Node](#)
 - [MS Association Rules Expert Options](#)
 - [MS Time Series Model Options](#)
 - [MS Time Series Settings Options](#)
 - [MS Sequence Clustering Node](#)
 - [MS Sequence Clustering Fields Options](#)
 - [MS Sequence Clustering Expert Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Time Series Settings Options

Begin Estimation. Specify the time period where you want predictions to start.

- Start From: New Prediction. The time period at which you want future predictions to start, expressed as an offset from the last time period of your historical data. For example, if your historical data ended at 12/99 and you wanted to begin predictions at 01/00, you would use a value of 1; however, if you wanted predictions to start at 03/00, you would use a value of 3.
- Start From: Historical Prediction. The time period at which you want historical predictions to start, expressed as a negative offset from the last time period of your historical data. For example, if your historical data ended at 12/99 and you wanted to make historical predictions for the last five time periods of your data, you would use a value of -5.

End Estimation. Specify the time period where you want predictions to stop.

- End step of the prediction. The time period at which you want predictions to stop, expressed as an offset from the last time period of your historical data. For example, if your historical data end at 12/99 and you want predictions to stop at 6/00, you would use a value of 6 here. For future predictions, the value must always be greater than or equal to the Start From value.

Related information

- [Building Models with Analysis Services](#)

- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Model Options](#)
- [MS Time Series Expert Options](#)
- [MS Sequence Clustering Node](#)
- [MS Sequence Clustering Fields Options](#)
- [MS Sequence Clustering Expert Options](#)
- [Settings Common to All Analysis Services Models](#)
- [MS Time Series Model Nugget](#)
- [MS Time Series Model Nugget Server Tab](#)
- [MS Time Series Model Nugget Settings Tab](#)
- [MS Sequence Clustering Model Nugget](#)
- [Exporting Models and Generating Nodes](#)

MS Sequence Clustering Node

The MS Sequence Clustering node uses a sequence analysis algorithm that explores data containing events that can be linked by following paths, or *sequences*. Some examples of this might be the click paths created when users navigate or browse a Web site, or the order in which a customer adds items to a shopping cart at an online retailer. The algorithm finds the most common sequences by grouping, or *clustering*, sequences that are identical.

Requirements

The requirements for a Microsoft Sequence Clustering model are:

- **ID field.** The Microsoft Sequence Clustering algorithm requires the sequence information to be stored in transactional format. For this, an ID field that identifies each transaction is required.
- **At least one input field.** The algorithm requires at least one input field.
- **Sequence field.** The algorithm also requires a sequence identifier field, which must have a measurement level of Continuous. For example, you can use a Web page identifier, an integer, or a text string, as long as the field identifies the events in a sequence. Only one sequence identifier is allowed for each sequence, and only one type of sequence is allowed in each model. The Sequence field must be different from the ID and Unique fields.
- **Target field.** A target field is required when building a sequence clustering model.
- **Unique field.** A sequence clustering model requires a key field that uniquely identifies records. You can set the Unique field to be the same as the ID field.
- [MS Sequence Clustering Fields Options](#)
- [MS Sequence Clustering Expert Options](#)

Related information

- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Node](#)
- [MS Time Series Model Options](#)
- [MS Time Series Expert Options](#)
- [MS Time Series Settings Options](#)
- [Settings Common to All Analysis Services Models](#)
- [MS Time Series Model Nugget](#)
- [MS Time Series Model Nugget Server Tab](#)
- [MS Time Series Model Nugget Settings Tab](#)
- [MS Sequence Clustering Model Nugget](#)

- [Exporting Models and Generating Nodes](#)
-

MS Sequence Clustering Fields Options

All modeling nodes have a Fields tab, where you specify the fields to be used in building the model.

Before you can build a sequence clustering model, you need to specify which fields you want to use as targets and as inputs. Note that for the MS Sequence Clustering node, you cannot use field information from an upstream Type node; you must specify the field settings here.

ID. Select an ID field from the list. Numeric or symbolic fields can be used as the ID field. Each unique value of this field should indicate a specific unit of analysis. For example, in a market basket application, each ID might represent a single customer. For a Web log analysis application, each ID might represent a computer (by IP address) or a user (by login data).

Inputs. Select the input field or fields for the model. These are the fields that contain the events of interest in sequence modeling.

Sequence. Choose a field from the list, to be used as the sequence identifier field. For example, you can use a Web page identifier, an integer, or a text string, provided that the field identifies the events in a sequence. Only one sequence identifier is allowed for each sequence, and only one type of sequence is allowed in each model. The Sequence field must be different from the ID field (specified on this tab) and the Unique field (specified on the Model tab).

Target. Choose a field to be used as the target field, that is, the field whose value you are trying to predict based on the sequence data.

Related information

- [Building Models with Analysis Services](#)
 - [Managing Analysis Services Models](#)
 - [MS Decision Tree Expert Options](#)
 - [MS Clustering Expert Options](#)
 - [MS Naive Bayes Expert Options](#)
 - [MS Linear Regression Expert Options](#)
 - [MS Neural Network Expert Options](#)
 - [MS Logistic Regression Expert Options](#)
 - [MS Association Rules Node](#)
 - [MS Association Rules Expert Options](#)
 - [MS Time Series Node](#)
 - [MS Time Series Model Options](#)
 - [MS Time Series Expert Options](#)
 - [MS Time Series Settings Options](#)
 - [MS Sequence Clustering Expert Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Sequence Clustering Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

Related information

- [Building Models with Analysis Services](#)
- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)

- [MS Time Series Node](#)
 - [MS Time Series Model Options](#)
 - [MS Time Series Expert Options](#)
 - [MS Time Series Settings Options](#)
 - [MS Sequence Clustering Fields Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

Scoring Analysis Services Models

Model scoring occurs within SQL Server and is performed by Analysis Services. The dataset may need to be uploaded to a temporary table if the data originates within IBM® SPSS® Modeler or needs to be prepared within IBM SPSS Modeler. Models that you create from IBM SPSS Modeler, using in-database mining, are actually a remote model held on the remote data mining or database server. This is an important distinction to understand as you browse and score models created using Microsoft Analysis Services algorithms.

In IBM SPSS Modeler, generally only a single prediction and associated probability or confidence is delivered.

For model scoring examples, see [Analysis Services Mining Examples](#).

- [Settings Common to All Analysis Services Models](#)
- [MS Time Series Model Nugget](#)
- [MS Sequence Clustering Model Nugget](#)
- [Exporting Models and Generating Nodes](#)

Related information

- [IBM SPSS Modeler and Microsoft Analysis Services](#)
 - [Requirements for Integration with Microsoft Analysis Services](#)
 - [Enabling Integration with Analysis Services](#)
 - [Building Models with Analysis Services](#)
-

Settings Common to All Analysis Services Models

The following settings are common to all Analysis Services models.

- [Analysis Services Model Nugget Server Tab](#)
- [Analysis Services Model Nugget Summary Tab](#)

Related information

- [Analysis Services Model Nugget Server Tab](#)
- [Analysis Services Model Nugget Summary Tab](#)
- [Exporting Models and Generating Nodes](#)
- [Building Models with Analysis Services](#)
- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naïve Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Node](#)
- [MS Time Series Model Options](#)
- [MS Time Series Expert Options](#)
- [MS Time Series Settings Options](#)
- [MS Sequence Clustering Node](#)
- [MS Sequence Clustering Fields Options](#)

- [MS Sequence Clustering Expert Options](#)
 - [MS Time Series Model Nugget](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
-

Analysis Services Model Nugget Server Tab

The Server tab is used to specify connections for in-database mining. The tab also provides the unique model key. The key is randomly generated when the model is built and is stored both within the model in IBM® SPSS® Modeler and also within the description of the model object stored in the Analysis Services database.

On the Server tab, you can configure the Analysis server host and database and the SQL Server data source for the scoring operation. Options specified here overwrite those specified in the Helper Applications or Build Model dialog boxes in IBM SPSS Modeler. See the topic [Enabling Integration with Analysis Services](#) for more information.

Model GUID. The model key is shown here. The key is randomly generated when the model is built and is stored both within the model in IBM SPSS Modeler and also within the description of the model object stored in the Analysis Services database.

Check. Click this button to check the model key against the key in the model stored in the Analysis Services database. This allows you to verify that the model still exists within the Analysis server and indicates that the structure of the model has not changed.

Note: The Check button is available only for models added to the stream canvas in preparation for scoring. If the check fails, investigate whether the model has been deleted or replaced by a different model on the server.

View. Click for a graphical view of the decision tree model. The Decision Tree Viewer is shared by other decision tree algorithms in IBM SPSS Modeler and the functionality is identical.

Related information

- [Analysis Services Model Nugget Summary Tab](#)
 - [Exporting Models and Generating Nodes](#)
-

Analysis Services Model Nugget Summary Tab

The Summary tab of a model nugget displays information about the model itself (*Analysis*), fields used in the model (*Fields*), settings used when building the model (*Build Settings*), and model training (*Training Summary*).

When you first browse the node, the Summary tab results are collapsed. To see the results of interest, use the expander control to the left of an item to unfold it or click the Expand All button to show all results. To hide the results when you have finished viewing them, use the expander control to collapse the specific results you want to hide or click the Collapse All button to collapse all results.

Analysis. Displays information about the specific model. If you have executed an Analysis node attached to this model nugget, information from that analysis will also appear in this section.

Fields. Lists the fields used as the target and the inputs in building the model.

Build Settings. Contains information about the settings used in building the model.

Training Summary. Shows the type of model, the stream used to create it, the user who created it, when it was built, and the elapsed time for building the model.

Related information

- [Analysis Services Model Nugget Server Tab](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Time Series Model Nugget

The MS Time Series model produces scores for only for the predicted time periods, not for the historical data.

The following table shows the fields that are added to the model.

Table 1. Fields added to the model

Field Name	Description
\$M-field	Predicted value of <i>field</i>
\$Var-field	Computed variance of <i>field</i>
\$Stdev-field	Standard deviation of <i>field</i>

- [MS Time Series Model Nugget Server Tab](#)
- [MS Time Series Model Nugget Settings Tab](#)

Related information

- [Building Models with Analysis Services](#)
- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Node](#)
- [MS Time Series Model Options](#)
- [MS Time Series Expert Options](#)
- [MS Time Series Settings Options](#)
- [MS Sequence Clustering Node](#)
- [MS Sequence Clustering Fields Options](#)
- [MS Sequence Clustering Expert Options](#)
- [Settings Common to All Analysis Services Models](#)
- [MS Sequence Clustering Model Nugget](#)
- [Exporting Models and Generating Nodes](#)

MS Time Series Model Nugget Server Tab

The Server tab is used to specify connections for in-database mining. The tab also provides the unique model key. The key is randomly generated when the model is built and is stored both within the model in IBM® SPSS® Modeler and also within the description of the model object stored in the Analysis Services database.

On the Server tab, you can configure the Analysis server host and database and the SQL Server data source for the scoring operation. Options specified here overwrite those specified in the Helper Applications or Build Model dialog boxes in IBM SPSS Modeler. See the topic [Enabling Integration with Analysis Services](#) for more information.

Model GUID. The model key is shown here. The key is randomly generated when the model is built and is stored both within the model in IBM SPSS Modeler and also within the description of the model object stored in the Analysis Services database.

Check. Click this button to check the model key against the key in the model stored in the Analysis Services database. This allows you to verify that the model still exists within the Analysis server and indicates that the structure of the model has not changed.

Note: The Check button is available only for models added to the stream canvas in preparation for scoring. If the check fails, investigate whether the model has been deleted or replaced by a different model on the server.

View. Click for a graphical view of the time series model. Analysis Services displays the completed model as a tree. You can also view a graph that shows the historical value of the target field over time, together with predicted future values.

For more information, see the description of the Time Series viewer in the MSDN library at <http://msdn.microsoft.com/en-us/library/ms175331.aspx>.

Related information

- [Building Models with Analysis Services](#)
- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)

- [MS Logistic Regression Expert Options](#)
 - [MS Association Rules Node](#)
 - [MS Association Rules Expert Options](#)
 - [MS Time Series Node](#)
 - [MS Time Series Model Options](#)
 - [MS Time Series Expert Options](#)
 - [MS Time Series Settings Options](#)
 - [MS Sequence Clustering Node](#)
 - [MS Sequence Clustering Fields Options](#)
 - [MS Sequence Clustering Expert Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget Settings Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Time Series Model Nugget Settings Tab

Begin Estimation. Specify the time period where you want predictions to start.

- Start From: New Prediction. The time period at which you want future predictions to start, expressed as an offset from the last time period of your historical data. For example, if your historical data ended at 12/99 and you wanted to begin predictions at 01/00, you would use a value of 1; however, if you wanted predictions to start at 03/00, you would use a value of 3.
- Start From: Historical Prediction. The time period at which you want historical predictions to start, expressed as a negative offset from the last time period of your historical data. For example, if your historical data ended at 12/99 and you wanted to make historical predictions for the last five time periods of your data, you would use a value of -5.

End Estimation. Specify the time period where you want predictions to stop.

- End step of the prediction. The time period at which you want predictions to stop, expressed as an offset from the last time period of your historical data. For example, if your historical data end at 12/99 and you want predictions to stop at 6/00, you would use a value of 6 here. For future predictions, the value must always be greater than or equal to the Start From value.

Related information

- [Building Models with Analysis Services](#)
 - [Managing Analysis Services Models](#)
 - [MS Decision Tree Expert Options](#)
 - [MS Clustering Expert Options](#)
 - [MS Naive Bayes Expert Options](#)
 - [MS Linear Regression Expert Options](#)
 - [MS Neural Network Expert Options](#)
 - [MS Logistic Regression Expert Options](#)
 - [MS Association Rules Node](#)
 - [MS Association Rules Expert Options](#)
 - [MS Time Series Node](#)
 - [MS Time Series Model Options](#)
 - [MS Time Series Expert Options](#)
 - [MS Time Series Settings Options](#)
 - [MS Sequence Clustering Node](#)
 - [MS Sequence Clustering Fields Options](#)
 - [MS Sequence Clustering Expert Options](#)
 - [Settings Common to All Analysis Services Models](#)
 - [MS Time Series Model Nugget Server Tab](#)
 - [MS Sequence Clustering Model Nugget](#)
 - [Exporting Models and Generating Nodes](#)
-

MS Sequence Clustering Model Nugget

The following table shows the fields that are added to the MS Sequence Clustering model (where *field* is the name of the target field).

Table 1. Fields added to the model

Field Name	Description
\$MC-field	Prediction of the cluster to which this sequence belongs.

Field Name	Description
\$MCP-field	Probability that this sequence belongs to the predicted cluster.
\$MS-field	Predicted value of <i>field</i>
\$MSP-field	Probability that \$MS-field value is correct.

Related information

- [Building Models with Analysis Services](#)
- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Node](#)
- [MS Time Series Model Options](#)
- [MS Time Series Expert Options](#)
- [MS Time Series Settings Options](#)
- [MS Sequence Clustering Node](#)
- [MS Sequence Clustering Fields Options](#)
- [MS Sequence Clustering Expert Options](#)
- [Settings Common to All Analysis Services Models](#)
- [MS Time Series Model Nugget](#)
- [MS Time Series Model Nugget Server Tab](#)
- [MS Time Series Model Nugget Settings Tab](#)
- [Exporting Models and Generating Nodes](#)

Exporting Models and Generating Nodes

You can export a model summary and structure to text and HTML format files. You can generate the appropriate Select and Filter nodes where appropriate.

Similar to other model nuggets in IBM® SPSS® Modeler, the Microsoft Analysis Services model nuggets support the direct generation of record and field operations nodes. Using the model nugget Generate menu options, you can generate the following nodes:

- Select node (only if an item is selected on the Model tab)
- Filter node

Related information

- [Building Models with Analysis Services](#)
- [Managing Analysis Services Models](#)
- [MS Decision Tree Expert Options](#)
- [MS Clustering Expert Options](#)
- [MS Naive Bayes Expert Options](#)
- [MS Linear Regression Expert Options](#)
- [MS Neural Network Expert Options](#)
- [MS Logistic Regression Expert Options](#)
- [MS Association Rules Node](#)
- [MS Association Rules Expert Options](#)
- [MS Time Series Node](#)
- [MS Time Series Model Options](#)
- [MS Time Series Expert Options](#)
- [MS Time Series Settings Options](#)
- [MS Sequence Clustering Node](#)
- [MS Sequence Clustering Fields Options](#)
- [MS Sequence Clustering Expert Options](#)
- [Settings Common to All Analysis Services Models](#)
- [MS Time Series Model Nugget](#)
- [MS Time Series Model Nugget Server Tab](#)
- [MS Time Series Model Nugget Settings Tab](#)

- [MS Sequence Clustering Model Nugget](#)
 - [Analysis Services Model Nugget Server Tab](#)
 - [Analysis Services Model Nugget Summary Tab](#)
-

Analysis Services Mining Examples

Included are a number of sample streams that demonstrate the use of MS Analysis Services data mining with IBM® SPSS® Modeler. These streams can be found in the IBM SPSS Modeler installation folder under:

|Demos|Database_Modelling|Microsoft

Note: The Demos folder can be accessed from the IBM SPSS Modeler program group on the Windows Start menu.

- [Example Streams: Decision Trees](#)

Related information

- [Example Streams: Decision Trees](#)
 - [Example Stream: Upload Data](#)
 - [Example Stream: Explore Data](#)
 - [Example Stream: Build Model](#)
 - [Example Stream: Evaluate Model](#)
 - [Example Stream: Deploy Model](#)
 - [Model Consistency](#)
-

Example Streams: Decision Trees

The following streams can be used together in sequence as an example of the database mining process using the Decision Trees algorithm provided by MS Analysis Services.

Table 1. Decision Trees - example streams

Stream	Description
<code>1_upload_data.str</code>	Used to clean and upload data from a flat file into the database.
<code>2_explore_data.str</code>	Provides an example of data exploration with IBM® SPSS® Modeler
<code>3_build_model.str</code>	Builds the model using the database-native algorithm.
<code>4_evaluate_model.str</code>	Used as an example of model evaluation with IBM SPSS Modeler
<code>5_deploy_model.str</code>	Deploys the model for in-database scoring.

Note: In order to run the example, streams must be executed in order. In addition, source and modeling nodes in each stream must be updated to reference a valid data source for the database you want to use.

The dataset used in the example streams concerns credit card applications and presents a classification problem with a mixture of categorical and continuous predictors. For more information about this dataset, see the `crx.names` file in the same folder as the sample streams.

This dataset is available from the UCI Machine Learning Repository at <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/credit-screening/>.

- [Example Stream: Upload Data](#)
- [Example Stream: Explore Data](#)
- [Example Stream: Build Model](#)
- [Example Stream: Evaluate Model](#)
- [Example Stream: Deploy Model](#)

Related information

- [Analysis Services Mining Examples](#)
- [Example Stream: Upload Data](#)
- [Example Stream: Explore Data](#)
- [Example Stream: Build Model](#)
- [Example Stream: Evaluate Model](#)
- [Example Stream: Deploy Model](#)

Example Stream: Upload Data

The first example stream, `1_upload_data.str`, is used to clean and upload data from a flat file into SQL Server.

Since Analysis Services data mining requires a key field, this initial stream uses a Derive node to add a new field to the dataset called `KEY` with unique values 1,2,3 using the IBM® SPSS® Modeler `@INDEX` function.

The subsequent Filler node is used for missing-value handling and replaces empty fields read in from the text file `crx.data` with `NULL` values.

Related information

- [Analysis Services Mining Examples](#)
- [Example Streams: Decision Trees](#)
- [Example Stream: Explore Data](#)
- [Example Stream: Build Model](#)
- [Example Stream: Evaluate Model](#)
- [Example Stream: Deploy Model](#)

Example Stream: Explore Data

The second example stream, `2_explore_data.str`, is used to demonstrate use of a Data Audit node to gain a general overview of the data, including summary statistics and graphs.

Double-clicking a graph in the Data Audit Report produces a more detailed graph for deeper exploration of a given field.

Related information

- [Analysis Services Mining Examples](#)
- [Example Streams: Decision Trees](#)
- [Example Stream: Upload Data](#)
- [Example Stream: Build Model](#)
- [Example Stream: Evaluate Model](#)
- [Example Stream: Deploy Model](#)

Example Stream: Build Model

The third example stream, `3_build_model.str`, illustrates model building in IBM® SPSS® Modeler. You can attach the database model to the stream and double-click to specify build settings.

On the Model tab of the dialog box, you can specify the following:

1. Select the Key field as the Unique ID field.

On the Expert tab, you can fine-tune settings for building the model.

Before running, ensure that you have specified the correct database for model building. Use the Server tab to adjust any settings.

Related information

- [Analysis Services Mining Examples](#)
- [Example Streams: Decision Trees](#)
- [Example Stream: Upload Data](#)
- [Example Stream: Explore Data](#)
- [Example Stream: Evaluate Model](#)
- [Example Stream: Deploy Model](#)

Example Stream: Evaluate Model

The fourth example stream, `4_evaluate_model.str`, illustrates the advantages of using IBM® SPSS® Modeler for in-database modeling. Once you have executed the model, you can add it back to your data stream and evaluate the model using several tools offered in IBM SPSS Modeler.

Viewing Modeling Results

You can double-click the model nugget to explore your results. The Summary tab provides a rule-tree view of your results. You can also click the View button, located on the Server tab, for a graphical view of the Decision Trees model.

Evaluating Model Results

The Analysis node in the sample stream creates a coincidence matrix showing the pattern of matches between each predicted field and its target field. Execute the Analysis node to view the results.

The Evaluation node in the sample stream can create a gains chart designed to show accuracy improvements made by the model. Execute the Evaluation node to view the results.

Related information

- [Analysis Services Mining Examples](#)
 - [Example Streams: Decision Trees](#)
 - [Example Stream: Upload Data](#)
 - [Example Stream: Explore Data](#)
 - [Example Stream: Build Model](#)
 - [Example Stream: Deploy Model](#)
-

Example Stream: Deploy Model

Once you are satisfied with the accuracy of the model, you can deploy it for use with external applications or for publishing back to the database. In the final example stream, `5_deploy_model.str`, data is read from the table `CREDIT` and then scored and published to the table `CREDITSCORES` using a Database Export node.

Running the stream generates the following SQL:

```
DROP TABLE CREDITSCORES

CREATE TABLE CREDITSCORES ( "field1" varchar(1), "field2" varchar(255), "field3" float, "field4" varchar(1), "field5" varchar(2), "field6" varchar(2), "field7" varchar(2), "field8" float, "field9" varchar(1), "field10" varchar(1), "field11" int, "field12" varchar(1), "field13" varchar(1), "field14" int, "field15" int, "field16" varchar(1), "KEY" int, "$M-field16" varchar(9), "$MC-field16" float )

INSERT INTO CREDITSCORES ("field1", "field2", "field3", "field4", "field5", "field6", "field7", "field8", "field9", "field10", "field11", "field12", "field13", "field14", "field15", "field16", "KEY", "$M-field16", "$MC-field16")
SELECT T0.C0 AS C0, T0.C1 AS C1, T0.C2 AS C2, T0.C3 AS C3, T0.C4 AS C4, T0.C5 AS C5,
       T0.C6 AS C6, T0.C7 AS C7, T0.C8 AS C8, T0.C9 AS C9, T0.C10 AS C10,
       T0.C11 AS C11, T0.C12 AS C12, T0.C13 AS C13, T0.C14 AS C14,
       T0.C15 AS C15, T0.C16 AS C16, T0.C17 AS C17, T0.C18 AS C18
FROM (
    SELECT CONVERT(NVARCHAR,[TA].[field1]) AS C0, CONVERT(NVARCHAR,[TA].[field2]) AS C1,
           [TA].[field3] AS C2, CONVERT(NVARCHAR,[TA].[field4]) AS C3,
           CONVERT(NVARCHAR,[TA].[field5]) AS C4, CONVERT(NVARCHAR,[TA].[field6]) AS C5,
           CONVERT(NVARCHAR,[TA].[field7]) AS C6, [TA].[field8] AS C7,
           CONVERT(NVARCHAR,[TA].[field9]) AS C8, CONVERT(NVARCHAR,[TA].[field10]) AS C9,
           [TA].[field11] AS C10, CONVERT(NVARCHAR,[TA].[field12]) AS C11,
           CONVERT(NVARCHAR,[TA].[field13]) AS C12, [TA].[field14] AS C13,
           [TA].[field15] AS C14, CONVERT(NVARCHAR,[TA].[field16]) AS C15,
           [TA].[KEY] AS C16, CONVERT(NVARCHAR,[TA].[$M-field16]) AS C17,
           [TA].[$MC-field16] AS C18
    FROM openrowset('MSOLAP',
        'Datasource=localhost;Initial catalog=FoodMart 2000',
        'SELECT [T].[C0] AS [field1], [T].[C1] AS [field2], [T].[C2] AS [field3],
               [T].[C3] AS [field4], [T].[C4] AS [field5], [T].[C5] AS [field6],
               [T].[C6] AS [field7], [T].[C7] AS [field8], [T].[C8] AS [field9],
               [T].[C9] AS [field10], [T].[C10] AS [field11], [T].[C11] AS [field12],
               [T].[C12] AS [field13], [T].[C13] AS [field14], [T].[C14] AS [field15],
               [T].[C15] AS [field16], [T].[C16] AS [KEY], [CREDIT1].[field16] AS [$M-field16],
               PredictProbability([CREDIT1].[field16]) AS [$MC-field16]
        FROM [CREDIT1] PREDICTION JOIN
              openrowset('MSDASQL',
                'Dsn=LocalServer;Uid=:pwd='', ''SELECT T0."field1" AS C0, T0."field2" AS C1,
                T0."field3" AS C2, T0."field4" AS C3, T0."field5" AS C4, T0."field6" AS C5,
                T0."field7" AS C6, T0."field8" AS C7, T0."field9" AS C8, T0."field10" AS C9,
                T0."field11" AS C10, T0."field12" AS C11, T0."field13" AS C12,
```

```

        T0."field14" AS C13,T0."field15" AS C14,T0."field16" AS C15,
        T0."KEY" AS C16 FROM "dbo".CREDITDATA TO'') AS [T]
    ON [T].[C2] = [CREDIT1].[field3] and [T].[C7] = [CREDIT1].[field8]
    and [T].[C8] = [CREDIT1].[field9] and [T].[C9] = [CREDIT1].[field10]
    and [T].[C10] = [CREDIT1].[field11] and [T].[C11] = [CREDIT1].[field12]
    and [T].[C14] = [CREDIT1].[field15]') AS [TA]
) TO

```

Related information

- [Analysis Services Mining Examples](#)
 - [Example Streams: Decision Trees](#)
 - [Example Stream: Upload Data](#)
 - [Example Stream: Explore Data](#)
 - [Example Stream: Build Model](#)
 - [Example Stream: Evaluate Model](#)
-

Database Modeling with Oracle Data Mining

- [About Oracle Data Mining](#)
 - [Requirements for Integration with Oracle](#)
 - [Enabling Integration with Oracle](#)
 - [Building Models with Oracle Data Mining](#)
 - [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Oracle Decision Tree](#)
 - [Oracle O-Cluster](#)
 - [Oracle k-Means](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Apriori](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [Managing Oracle Models](#)
 - [Preparing the Data](#)
 - [Oracle Data Mining Examples](#)
-

About Oracle Data Mining

IBM® SPSS® Modeler supports integration with Oracle Data Mining (ODM), which provides a family of data mining algorithms tightly embedded within the Oracle RDBMS. These features can be accessed through the IBM SPSS Modeler graphical user interface and workflow-oriented development environment, allowing customers to use the data mining algorithms offered by ODM.

IBM SPSS Modeler supports integration of the following algorithms from Oracle Data Mining:

- Naive Bayes
- Adaptive Bayes
- Support Vector Machine (SVM)
- Generalized Linear Models (GLM)*
- Decision Tree
- O-Cluster
- k-Means
- Nonnegative Matrix Factorization (NMF)
- Apriori
- Minimum Descriptor Length (MDL)
- Attribute Importance (AI)

* 11g R1 only

Related information

- [Managing Oracle Models](#)

- [Oracle Model Nugget Server Tab](#)
 - [Oracle Model Nugget Summary Tab](#)
 - [Oracle Model Nugget Settings Tab](#)
 - [Listing Oracle Models](#)
 - [Preparing the Data](#)
 - [Oracle Naïve Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Decision Tree](#)
 - [Oracle O-Cluster](#)
 - [Oracle k-Means](#)
 - [Oracle Apriori](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Data Mining Examples](#)
-

Requirements for Integration with Oracle

The following conditions are prerequisites for conducting in-database modeling using Oracle Data Mining. You may need to consult with your database administrator to ensure that these conditions are met.

- IBM® SPSS® Modeler running in local mode or against an IBM SPSS Modeler Server installation on Windows or UNIX.
- Oracle 10gR2 or 11gR1 (10.2 Database or higher) with the Oracle Data Mining option.

Note: 10gR2 provides support for all database modeling algorithms except Generalized Linear Models (requires 11gR1).

- An ODBC data source for connecting to Oracle. See the topic [Enabling Integration with Oracle](#) for more information.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > About > Additional Details

If connectivity is enabled, you see the option Server Enablement in the License Status tab.

See [Connecting to IBM SPSS Modeler Server](#) for more information.

Related information

- [Oracle Naïve Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle O-Cluster](#)
 - [Oracle k-Means](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Apriori](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Data Mining Examples](#)
-

Enabling Integration with Oracle

To enable the IBM® SPSS® Modeler integration with Oracle Data Mining, you'll need to configure Oracle, create an ODBC source, enable the integration in the IBM SPSS Modeler Helper Applications dialog box, and enable SQL generation and optimization.

Configuring Oracle

To install and configure Oracle Data Mining, see the Oracle documentation—in particular, the *Oracle Administrator's Guide*—for more details.

Creating an ODBC Source for Oracle

To enable the connection between Oracle and IBM SPSS Modeler, you need to create an ODBC system data source name (DSN).

Before creating a DSN, you should have a basic understanding of ODBC data sources and drivers, and database support in IBM SPSS Modeler.

If you are running in distributed mode against IBM SPSS Modeler Server, create the DSN on the server computer. If you are running in local (client) mode, create the DSN on the client computer.

1. Install the ODBC drivers. These are available on the IBM SPSS Data Access Pack installation disk shipped with this release. Run the *setup.exe* file to start the installer, and select all the relevant drivers. Follow the on-screen instructions to install the drivers.
 - a. Create the DSN.

Note: The menu sequence depends on your version of Windows.

 - Windows XP. From the Start menu, choose Control Panel. Double-click Administrative Tools, and then double-click Data Sources (ODBC).
 - Windows Vista. From the Start menu, choose Control Panel, then System Maintenance. Double-click Administrative Tools, selectData Sources (ODBC), then click Open.
 - Windows 7. From the Start menu, choose Control Panel, then System & Security, then Administrative Tools. SelectData Sources (ODBC), then click Open.
 - b. Go to the System DSN tab, and then click Add.
2. Select the SPSS OEM 6.0 Oracle Wire Protocol driver.
3. Click Finish.
4. In the ODBC Oracle Wire Protocol Driver Setup screen, enter a data source name of your choosing, the hostname of the Oracle server, the port number for the connection, and the SID for the Oracle instance you are using.

The hostname, port, and SID can be obtained from the *tnsnames.ora* file on the server machine if you have implemented TNS with a *tnsnames.ora* file. Contact your Oracle administrator for more information.
5. Click the Test button to test the connection.

Enabling Oracle Data Mining Integration in IBM SPSS Modeler

1. From the IBM SPSS Modeler menus choose:
Tools > *Options* > *Helper Applications*

2. Click the Oracle tab.

Enable Oracle Data Mining Integration. Enables the Database Modeling palette (if not already displayed) at the bottom of the IBM SPSS Modeler window and adds the nodes for Oracle Data Mining algorithms.

Oracle Connection. Specify the default Oracle ODBC data source used for building and storing models, along with a valid user name and password. This setting can be overridden on the individual modeling nodes and model nuggets.

Note: The database connection used for modeling purposes may or may not be the same as the connection used to access data. For example, you might have a stream that accesses data from one Oracle database, downloads the data to IBM SPSS Modeler for cleaning or other manipulations, and then uploads the data to a different Oracle database for modeling purposes. Alternatively, the original data might reside in a flat file or other (non-Oracle) source, in which case it would need to be uploaded to Oracle for modeling. In all cases, the data will be automatically uploaded to a temporary table created in the database that is used for modeling.

Warn when about to overwrite an Oracle Data Mining model. Select this option to ensure that models stored in the database are not overwritten by IBM SPSS Modeler without warning.

List Oracle Data Mining Models. Displays available data mining models.

Enable launch of Oracle Data Miner. (optional) When enabled, allows IBM SPSS Modeler to launch the Oracle Data Miner application. Refer to [Oracle Data Miner](#) for more information.

Path for Oracle Data Miner executable. (optional) Specifies the physical location of the Oracle Data Miner for Windows executable file (for example, C:\odm\bin\odminerw.exe). Oracle Data Miner is not installed with IBM SPSS Modeler; you must download the correct version from the Oracle Web site (<http://www.oracle.com/technology/products/bi/odm/odminer.html>) and install it at the client.

Enabling SQL Generation and Optimization

1. From the IBM SPSS Modeler menus choose:
Tools > *Stream Properties* > *Options*
2. Click the Optimization option in the navigation pane.
3. Confirm that the Generate SQL option is enabled. This setting is required for database modeling to function.
4. Select Optimize SQL Generation and Optimize other execution (not strictly required but strongly recommended for optimized performance).

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)

- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Data Mining Examples](#)

Building Models with Oracle Data Mining

Oracle model-building nodes work just like other modeling nodes in IBM® SPSS® Modeler, with a few exceptions. You can access these nodes from the Database Modeling palette across the bottom of the IBM SPSS Modeler window.

Data Considerations

Oracle requires that categorical data be stored in a string format (either CHAR or VARCHAR2). As a result, IBM SPSS Modeler will not allow numeric storage fields with a measurement level of *Flag* or *Nominal* (categorical) to be specified as input to ODM models. If necessary, numbers can be converted to strings in IBM SPSS Modeler by using the Reclassify node.

Target field. Only one field may be selected as the output (target) field in ODM classification models.

Model name. From Oracle 11gR1 onwards, the name `unique` is a keyword and cannot be used as a custom model name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

General Comments

- PMML Export/Import is not provided from IBM SPSS Modeler for models created by Oracle Data Mining.
- Model scoring always happens within ODM. The dataset may need to be uploaded to a temporary table if the data originate, or need to be prepared, within IBM SPSS Modeler.
- In IBM SPSS Modeler, generally only a single prediction and associated probability or confidence is delivered.
- IBM SPSS Modeler restricts the number of fields that can be used in model building and scoring to 1,000.
- IBM SPSS Modeler can score ODM models from within streams published for execution by using IBM SPSS Modeler Solution Publisher.
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Data Mining Examples](#)

Oracle Models Server Options

Specify the Oracle connection that is used to upload data for modeling. If necessary, you can select a connection on the Server tab for each modeling node to override the default Oracle connection specified in the Helper Applications dialog box. See the topic [Enabling Integration with Oracle](#) for more information.

Comments

- The connection used for modeling may or may not be the same as the connection used in the source node for a stream. For example, you might have a stream that accesses data from one Oracle database, downloads the data to IBM® SPSS® Modeler for cleaning or other manipulations, and then uploads the data to a different Oracle database for modeling purposes.
- The ODBC data source name is effectively embedded in each IBM SPSS Modeler stream. If a stream that is created on one host is executed on a different host, the name of the data source must be the same on each host. Alternatively, a different data source can be selected on the Server tab in each source or modeling node.

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Decision Tree](#)
 - [Oracle O-Cluster](#)
 - [Oracle k-Means](#)
 - [Oracle Apriori](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Misclassification Costs](#)
 - [Naive Bayes Model Options](#)
 - [Naive Bayes Expert Options](#)
 - [Adaptive Bayes Model Options](#)
 - [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [K-Means Model Options](#)
 - [K-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
-

Misclassification Costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

With the exception of C5.0 models, misclassification costs are not applied when scoring a model and are not taken into account when ranking or comparing models using an Auto Classifier node, evaluation chart, or Analysis node. A model that includes costs may not produce fewer errors than one that doesn't and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of *less expensive* errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select Use misclassification costs and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying *A* as *B* to be 2.0, the cost of misclassifying *B* as *A* will still have the default value of 1.0 unless you explicitly change it as well.

Note: Only the Decision Trees model allows costs to be specified at build time.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Models Server Options](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)

- [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Oracle Decision Tree](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [Oracle O-Cluster](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [Oracle k-Means](#)
 - [k-Means Model Options](#)
 - [k-Means Expert Options](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Oracle Apriori](#)
 - [Apriori Model Options](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
-

Oracle Naive Bayes

Naive Bayes is a well-known algorithm for classification problems. The model is termed *naïve* because it treats all proposed prediction variables as being independent of one another. Naive Bayes is a fast, scalable algorithm that calculates conditional probabilities for combinations of attributes and the target attribute. From the training data, an independent probability is established. This probability gives the likelihood of each target class, given the occurrence of each value category from each input variable.

- Cross-validation is used to test model accuracy on the same data that were used to build the model. This is particularly useful when the number of cases available to build a model is small.
- The model output can be browsed in a matrix format. The numbers in the matrix are conditional probabilities that relate the predicted classes (columns) and predictor variable-value combinations (rows).
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)

Related information

- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [k-Means Model Options](#)

- [K-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
 - [Oracle Data Mining Examples](#)
-

Naive Bayes Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM® SPSS® Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11g only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Decision Tree](#)
 - [Oracle O-Cluster](#)
 - [Oracle k-Means](#)
 - [Oracle Apriori](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Models Server Options](#)
 - [Misclassification Costs](#)
 - [Naive Bayes Expert Options](#)
 - [Adaptive Bayes Model Options](#)
 - [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [k-Means Model Options](#)
 - [k-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
-

Naive Bayes Expert Options

When the model is built, individual predictor attribute values or value pairs are ignored unless there are enough occurrences of a given value or pair in the training data. The thresholds for ignoring values are specified as fractions based on the number of records in the training data.

Adjusting these thresholds may reduce noise and improve the model's ability to generalize to other datasets.

- **Singleton Threshold.** Specifies the threshold for a given predictor attribute value. The number of occurrences of a given value must equal or exceed the specified fraction or the value is ignored.
- **Pairwise Threshold.** Specifies the threshold for a given attribute and predictor value pair. The number of occurrences of a given value pair must equal or exceed the specified fraction or the pair is ignored.

Prediction probability. Allows the model to include the probability of a correct prediction for a possible outcome of the target field. To enable this feature, choose Select, click the Specify button, choose one of the possible outcomes, and then click Insert.

Use Prediction Set. Generates a table of all the possible results for all possible outcomes of the target field.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)
- [AI Selection Options](#)

Oracle Adaptive Bayes

Adaptive Bayes Network (ABN) constructs Bayesian Network classifiers by using Minimum Description Length (MDL) and automatic feature selection. ABN does well in certain situations where Naive Bayes does poorly and does at least as well in most other situations, although performance may be slower. The ABN algorithm provides the ability to build three types of advanced, Bayesian-based models, including simplified decision tree (single-feature), pruned Naive Bayes, and boosted multifeature models.

Note: The Oracle Adaptive Bayes algorithm has been dropped in Oracle 12C and is not supported in IBM® SPSS® Modeler when using Oracle 12C. See http://docs.oracle.com/database/121/DMPRG/release_changes.htm#DMPRG726.

Generated Models

In single-feature build mode, ABN produces a simplified decision tree, based on a set of human-readable rules, that allows the business user or analyst to understand the basis of the model's predictions and act or explain to others accordingly. This may be a significant advantage over Naive Bayes and multifeature models. These rules can be browsed like a standard rule set in IBM SPSS Modeler. A simple set of rules might look like this:

```
IF MARITAL_STATUS = "Married"  
AND EDUCATION_NUM = "13-16"
```

```
THEN CHURN= "TRUE"
Confidence = .78, Support = 570 cases
```

Pruned Naive Bayes and multifeature models cannot be browsed in IBM SPSS Modeler.

- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)

Related information

- [Oracle Naive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)
- [AI Selection Options](#)
- [Oracle Data Mining Examples](#)

Adaptive Bayes Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM® SPSS® Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Model Type

You can choose from three different modes for building the model.

- **Multi-feature.** Builds and compares a number of models, including an NB model and single and multifeature product probability models. This is the most exhaustive mode and typically takes the longest to compute as a result. Rules are produced only if the single feature model turns out to be best. If a multifeature or NB model is chosen, no rules are produced.
- **Single-feature.** Creates a simplified decision tree based on a set of rules. Each rule contains a condition together with probabilities associated with each outcome. The rules are mutually exclusive and are provided in a format that can be read by humans, which may be a significant advantage over Naive Bayes and multifeature models.
- **Naive Bayes.** Builds a single NB model and compares it with the global sample prior (the distribution of target values in the global sample). The NB model is produced as output only if it turns out to be a better predictor of the target values than the global prior. Otherwise, no model is produced as output.

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Decision Tree](#)
 - [Oracle O-Cluster](#)
 - [Oracle k-Means](#)
 - [Oracle Apriori](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Models Server Options](#)
 - [Misclassification Costs](#)
 - [Naive Bayes Model Options](#)
 - [Naive Bayes Expert Options](#)
 - [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [K-Means Model Options](#)
 - [K-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
-

Adaptive Bayes Expert Options

Limit execution time. Select this option to specify a maximum build time in minutes. This makes it possible to produce models in less time, although the resulting model may be less accurate. At each milestone in the modeling process, the algorithm checks whether it will be able to complete the next milestone within the specified amount of time before continuing and returns the best model available when the limit is reached.

Max Predictors. This option allows you to limit the complexity of the model and improve performance by limiting the number of predictors used. Predictors are ranked based on an MDL measure of their correlation to the target as a measure of their likelihood of being included in the model.

Max Naive Bayes Predictors. This option specifies the maximum number of predictors to be used in the Naive Bayes model.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)

- [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [K-Means Model Options](#)
 - [K-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
 - [Oracle Data Mining Examples](#)
-

Oracle Support Vector Machine (SVM)

Support Vector Machine (SVM) is a classification and regression algorithm that uses machine learning theory to maximize predictive accuracy without overfitting the data. SVM uses an optional nonlinear transformation of the training data, followed by the search for regression equations in the transformed data to separate the classes (for categorical targets) or fit the target (for continuous targets). Oracle's implementation of SVM allows models to be built by using one of two available kernels, linear or Gaussian. The linear kernel omits the nonlinear transformation altogether so that the resulting model is essentially a regression model.

For more information, see the *Oracle Data Mining Application Developer's Guide* and *Oracle Data Mining Concepts*.

- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)
- [AI Selection Options](#)
- [Oracle Data Mining Examples](#)

Oracle SVM Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM® SPSS® Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11g only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Active Learning. Provides a way to deal with large build sets. With active learning, the algorithm creates an initial model based on a small sample before applying it to the complete training dataset and then incrementally updates the sample and model based on the results. The cycle is repeated until the model converges on the training data or the maximum allowed number of support vectors is reached.

Kernel Function. Select Linear or Gaussian, or leave the default System Determined to allow the system to choose the most suitable kernel. Gaussian kernels are able to learn more complex relationships but generally take longer to compute. You may want to start with the linear kernel and try the Gaussian kernel only if the linear kernel fails to find a good fit. This is more likely to happen with a regression model, where the choice of kernel matters more. Also, note that SVM models built with the Gaussian kernel cannot be browsed in IBM SPSS Modeler. Models built with the linear kernel can be browsed in IBM SPSS Modeler in the same manner as standard regression models.

Normalization Method. Specifies the normalization method for continuous input and target fields. You can choose Z-Score, Min-Max, or None. Oracle performs normalization automatically if the Auto Data Preparation check box is selected. Uncheck this box to select the normalization method manually.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [k-Means Model Options](#)
- [k-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)
- [AI Selection Options](#)
- [Oracle Data Mining Examples](#)

Oracle SVM Expert Options

Kernel Cache Size. Specifies, in bytes, the size of the cache to be used for storing computed kernels during the build operation. As might be expected, larger caches typically result in faster builds. The default is 50MB.

Convergence Tolerance. Specifies the tolerance value that is allowed before termination for the model build. The value must be between 0 and 1. The default value is 0.001. Larger values tend to result in faster building but less-accurate models.

Specify Standard Deviation. Specifies the standard deviation parameter used by the Gaussian kernel. This parameter affects the trade-off between model complexity and ability to generalize to other datasets (overfitting and underfitting the data). Higher standard deviation values favor underfitting. By default, this parameter is estimated from the training data.

Specify Epsilon. For regression models only, specifies the value of the interval of the allowed error in building epsilon-insensitive models. In other words, it distinguishes small errors (which are ignored) from large errors (which aren't). The value must be between 0 and 1. By default, this is estimated from the training data.

Specify Complexity Factor. Specifies the complexity factor, which trades off model error (as measured against the training data) and model complexity in order to avoid overfitting or underfitting the data. Higher values place a greater penalty on errors, with an increased risk of overfitting the data; lower values place a lower penalty on errors and can lead to underfitting.

Specify Outlier Rate. Specifies the desired rate of outliers in the training data. Only valid for One-Class SVM models. Cannot be used with the **Specify Complexity Factor** setting.

Prediction probability. Allows the model to include the probability of a correct prediction for a possible outcome of the target field. To enable this feature, choose Select, click the Specify button, choose one of the possible outcomes, and then click Insert.

Use Prediction Set. Generates a table of all the possible results for all possible outcomes of the target field.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)
- [AI Selection Options](#)
- [Oracle Data Mining Examples](#)

Oracle SVM Weights Options

In a classification model, using weights enables you to specify the relative importance of the various possible target values. Doing so might be useful, for example, if the data points in your training data are not realistically distributed among the categories. Weights enable you to bias the model so that you can compensate for those categories that are less well represented in the data. Increasing the weight for a target value should increase the percentage of correct predictions for that category.

There are three methods of setting weights:

- Based on training data. This is the default. Weights are based on the relative frequencies of the categories in the training data.
- Equal for all classes. Weights for all categories are defined as $1/k$, where k is the number of target categories.
- Custom. You can specify your own weights. Starting values for weights are set as equal for all classes. You can adjust the weights for individual categories to user-defined values. To adjust a specific category's weight, select the Weight cell in the table corresponding to the desired category, delete the contents of the cell, and enter the desired value.

The weights for all categories should sum to 1.0. If they do not sum to 1.0, a warning is displayed, with an option to automatically normalize the values. This automatic adjustment preserves the proportions across categories while enforcing the weight constraint. You can perform this adjustment at any time by clicking the Normalize button. To reset the table to equal values for all categories, click the Equalize button.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AT Model Options](#)
- [AT Selection Options](#)
- [Oracle Data Mining Examples](#)

Oracle Generalized Linear Models (GLM)

(11g only) Generalized linear models relax the restrictive assumptions made by linear models. These include, for example, the assumptions that the target variable has a normal distribution, and that the effect of the predictors on the target variable is linear in nature. A generalized linear model is suitable for predictions where the distribution of the target is likely to have a non-normal distribution, such as a multinomial or a Poisson distribution. Similarly, a generalized linear model is useful in cases where the relationship, or link, between the predictors and the target is likely to be non-linear.

For more information, see the *Oracle Data Mining Application Developer's Guide* and *Oracle Data Mining Concepts*.

- [Oracle GLM Model Options](#)
- [Oracle GLM Expert Options](#)
- [Oracle GLM Weights Options](#)

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)

- [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Decision Tree](#)
 - [Oracle O-Cluster](#)
 - [Oracle k-Means](#)
 - [Oracle Apriori](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Models Server Options](#)
 - [Misclassification Costs](#)
 - [Naive Bayes Model Options](#)
 - [Naive Bayes Expert Options](#)
 - [Adaptive Bayes Model Options](#)
 - [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [K-Means Model Options](#)
 - [K-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
 - [Oracle Data Mining Examples](#)
-

Oracle GLM Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM® SPSS® Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11g only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Normalization Method. Specifies the normalization method for continuous input and target fields. You can choose Z-Score, Min-Max, or None. Oracle performs normalization automatically if the Auto Data Preparation check box is selected. Uncheck this box to select the normalization method manually.

Missing Value Handling. Specifies how to process missing values in the input data:

- Replace with mean or mode replaces missing values of numerical attributes with the mean value, and replaces missing values of categorical attributes with the mode.
 - Only use complete records ignores records with missing values.
-

Oracle GLM Expert Options

Use Row Weights. Check this box to activate the adjacent drop-down list, from where you can select a column that contains a weighting factor for the rows.

Save Row Diagnostics to Table. Check this box to activate the adjacent text field, where you can enter the name of a table to contain row-level diagnostics.

Coefficient Confidence Level. The degree of certainty, from 0.0 to 1.0, that the value predicted for the target will lie within a confidence interval computed by the model. Confidence bounds are returned with the coefficient statistics.

Reference Category for Target. Select Custom to choose a value for the target field to use as a reference category, or leave the default value Auto.

Ridge Regression. Ridge regression is a technique that compensates for the situation where there is too high a degree of correlation in the variables. You can use the Auto option to allow the algorithm to control the use of this technique, or you can control it manually by means of the Disable and Enable options. If you choose to enable ridge regression manually, you can override the system default value for the ridge parameter by entering a value in the adjacent field.

Produce VIF for Ridge Regression. Check this box if you want to produce Variance Inflation Factor (VIF) statistics when ridge is being used for linear regression.

Prediction probability. Allows the model to include the probability of a correct prediction for a possible outcome of the target field. To enable this feature, choose Select, click the Specify button, choose one of the possible outcomes, and then click Insert.

Use Prediction Set. Generates a table of all the possible results for all possible outcomes of the target field.

Oracle GLM Weights Options

In a classification model, using weights enables you to specify the relative importance of the various possible target values. Doing so might be useful, for example, if the data points in your training data are not realistically distributed among the categories. Weights enable you to bias the model so that you can compensate for those categories that are less well represented in the data. Increasing the weight for a target value should increase the percentage of correct predictions for that category.

There are three methods of setting weights:

- Based on training data. This is the default. Weights are based on the relative frequencies of the categories in the training data.
- Equal for all classes. Weights for all categories are defined as $1/k$, where k is the number of target categories.
- Custom. You can specify your own weights. Starting values for weights are set as equal for all classes. You can adjust the weights for individual categories to user-defined values. To adjust a specific category's weight, select the Weight cell in the table corresponding to the desired category, delete the contents of the cell, and enter the desired value.

The weights for all categories should sum to 1.0. If they do not sum to 1.0, a warning is displayed, with an option to automatically normalize the values. This automatic adjustment preserves the proportions across categories while enforcing the weight constraint. You can perform this adjustment at any time by clicking the Normalize button. To reset the table to equal values for all categories, click the Equalize button.

Oracle Decision Tree

Oracle Data Mining offers a classic Decision Tree feature, based on the popular Classification and Regression Tree algorithm. The ODM Decision Tree model contains complete information about each node, including Confidence, Support, and Splitting Criterion. The full Rule for each node can be displayed, and in addition, a surrogate attribute is supplied for each node, to be used as a substitute when applying the model to a case with missing values.

Decision trees are popular because they are so universally applicable, easy to apply and easy to understand. Decision trees sift through each potential input attribute searching for the best “splitter,” that is, attribute cutpoint (`AGE > 55`, for example) that splits the downstream data records into more homogeneous populations. After each split decision, ODM repeats the process growing out the entire tree and creating terminal “leaves” that represent similar populations of records, items, or people. Looking down from the root tree node (for example, the total population), decision trees provide human readable rules of `IF A, then B` statements. These decision tree rules also provide the support and confidence for each tree node.

While Adaptive Bayes Networks can also provide short simple rules that can be useful in providing explanations for each prediction, Decision Trees provide full Oracle Data Mining rules for each splitting decision. Decision Trees are also useful for developing detailed profiles of the best customers, healthy patients, factors associated with fraud, and so on.

- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)

- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Models Server Options](#)
 - [Misclassification Costs](#)
 - [Naive Bayes Model Options](#)
 - [Naive Bayes Expert Options](#)
 - [Adaptive Bayes Model Options](#)
 - [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [K-Means Model Options](#)
 - [K-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
 - [Oracle Data Mining Examples](#)
-

Decision Tree Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM® SPSS® Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11g only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Impurity metric. Specifies which metric is used for seeking the best test question for splitting data at each node. The best splitter and split value are those that result in the largest increase in target value homogeneity for the entities in the node. Homogeneity is measured in accordance with a metric. The supported metrics are **gini** and **entropy**.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Apriori](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle O-Cluster](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)

- [O-Cluster Expert Options](#)
 - [K-Means Model Options](#)
 - [K-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
-

Decision Tree Expert Options

Maximum Depth. Sets the maximum depth of the tree model to be built.

Minimum percentage of records in a node. Sets the percentage of the minimum number of records per node.

Minimum percentage of records for a split. Sets the minimum number of records in a parent node expressed as a percent of the total number of records used to train the model. No split is attempted if the number of records is below this percentage.

Minimum records in a node. Sets the minimum number of records returned.

Minimum records for a split. Sets the minimum number of records in a parent node expressed as a value. No split is attempted if the number of records is below this value.

Rule identifier. If checked, includes in the model a string to identify the node in the tree at which a particular split is made.

Prediction probability. Allows the model to include the probability of a correct prediction for a possible outcome of the target field. To enable this feature, choose Select, click the Specify button, choose one of the possible outcomes, and then click Insert.

Use Prediction Set. Generates a table of all the possible results for all possible outcomes of the target field.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)
- [AI Selection Options](#)
- [Oracle Data Mining Examples](#)

Oracle O-Cluster

The Oracle O-Cluster algorithm identifies naturally occurring groupings within a data population. Orthogonal partitioning clustering (O-Cluster) is an Oracle proprietary clustering algorithm that creates a hierarchical grid-based clustering model, that is, it creates axis-parallel (orthogonal) partitions in the input attribute space. The algorithm operates recursively. The resulting hierarchical structure represents an irregular grid that tessellates the attribute space into clusters.

The O-Cluster algorithm handles both numeric and categorical attributes and ODM will automatically select the best cluster definitions. ODM provides cluster detail information, cluster rules, cluster centroid values, and can be used to score a population on their cluster membership.

- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)
- [AI Selection Options](#)
- [Oracle Data Mining Examples](#)

O-Cluster Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM® SPSS® Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11g only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Maximum number of clusters. Sets the maximum number of generated clusters.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Apriori](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle O-Cluster](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)
- [AI Selection Options](#)

O-Cluster Expert Options

Maximum Buffer. Sets the maximum buffer size.

Sensitivity. Sets a fraction that specifies the peak density required for separating a new cluster. The fraction is related to the global uniform density.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)

- [k-Means Model Options](#)
 - [k-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
 - [Oracle Data Mining Examples](#)
-

Oracle k-Means

The Oracle k-Means algorithm identifies naturally occurring groupings within a data population. The k-Means algorithm is a distance-based clustering algorithm that partitions the data into a predetermined number of clusters (provided there are enough distinct cases). Distance-based algorithms rely on a distance metric (function) to measure the similarity between data points. Data points are assigned to the nearest cluster according to the distance metric used. ODM provides an enhanced version of k-Means.

The k-Means algorithm supports hierarchical clusters, handles numeric and categorical attributes, and cuts the population into the user-specified number of clusters. ODM provides cluster detail information, cluster rules, cluster centroid values, and can be used to score a population on their cluster membership.

- [k-Means Model Options](#)
- [k-Means Expert Options](#)

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Decision Tree](#)
 - [Oracle O-Cluster](#)
 - [Oracle Apriori](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Models Server Options](#)
 - [Misclassification Costs](#)
 - [Naive Bayes Model Options](#)
 - [Naive Bayes Expert Options](#)
 - [Adaptive Bayes Model Options](#)
 - [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [k-Means Model Options](#)
 - [k-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
 - [Oracle Data Mining Examples](#)
-

k-Means Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM® SPSS® Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11g only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Number of clusters. Sets the number of generated clusters

Distance Function. Specifies which distance function is used for k-Means Clustering.

Split criterion. Specifies which split criterion is used for k-Means Clustering.

Normalization Method. Specifies the normalization method for continuous input and target fields. You can choose Z-Score, Min-Max, or None.

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Apriori](#)
 - [Oracle Decision Tree](#)
 - [Oracle k-Means](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle O-Cluster](#)
 - [Oracle Models Server Options](#)
 - [Misclassification Costs](#)
 - [Naive Bayes Model Options](#)
 - [Naive Bayes Expert Options](#)
 - [Adaptive Bayes Model Options](#)
 - [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [k-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
-

k-Means Expert Options

Iterations. Sets the number of iterations for the k-Means algorithm.

Convergence tolerance. Sets the convergence tolerance for the k-Means algorithm.

Number of bins. Specifies the number of bins in the attribute histogram produced by k-Means. The bin boundaries for each attribute are computed globally on the entire training dataset. The binning method is equi-width. All attributes have the same number of bins with the exception of attributes with a single value that have only one bin.

Block growth. Sets the growth factor for memory allocated to hold cluster data.

Minimum Percent Attribute Support. Sets the fraction of attribute values that must be non-null in order for the attribute to be included in the rule description for the cluster. Setting the parameter value too high in data with missing values can result in very short or even empty rules.

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Decision Tree](#)
 - [Oracle O-Cluster](#)
 - [Oracle k-Means](#)
 - [Oracle Apriori](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Models Server Options](#)
 - [Misclassification Costs](#)
 - [Naive Bayes Model Options](#)
 - [Naive Bayes Expert Options](#)
 - [Adaptive Bayes Model Options](#)
 - [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [k-Means Model Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
 - [Oracle Data Mining Examples](#)
-

Oracle Nonnegative Matrix Factorization (NMF)

Nonnegative Matrix Factorization (NMF) is useful for reducing a large dataset into representative attributes. Similar to Principal Components Analysis (PCA) in concept, but able to handle larger amounts of attributes and in an additive representation model, NMF is a powerful, state-of-the-art data mining algorithm that can be used for a variety of use cases.

NMF can be used to reduce large amounts of data, text data for example, into smaller, more sparse representations that reduce the dimensionality of the data (the same information can be preserved using far fewer variables). The output of NMF models can be analyzed using supervised learning techniques such as SVMs or unsupervised learning techniques such as clustering techniques. Oracle Data Mining uses NMF and SVM algorithms to mine unstructured text data.

- [NMF Model Options](#)
- [NMF Expert Options](#)

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Apriori](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle O-Cluster](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)

- [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [K-Means Model Options](#)
 - [K-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
 - [Oracle Data Mining Examples](#)
-

NMF Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM® SPSS® Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11g only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Normalization Method. Specifies the normalization method for continuous input and target fields. You can choose Z-Score, Min-Max, or None. Oracle performs normalization automatically if the Auto Data Preparation check box is selected. Uncheck this box to select the normalization method manually.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Apriori](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle O-Cluster](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)

- [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
-

NMF Expert Options

Specify number of features. Specifies the number of features to be extracted.

Random seed. Sets the random seed for the NMF algorithm.

Number of iterations. Sets the number of iterations for the NMF algorithm.

Convergence tolerance. Sets the convergence tolerance for the NMF algorithm.

Display all features. Displays the feature ID and confidence for all features, instead of those values for only the best feature.

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Decision Tree](#)
 - [Oracle O-Cluster](#)
 - [Oracle k-Means](#)
 - [Oracle Apriori](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Models Server Options](#)
 - [Misclassification Costs](#)
 - [Naive Bayes Model Options](#)
 - [Naive Bayes Expert Options](#)
 - [Adaptive Bayes Model Options](#)
 - [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [K-Means Model Options](#)
 - [K-Means Expert Options](#)
 - [NMF Model Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
 - [Oracle Data Mining Examples](#)
-

Oracle Apriori

The Apriori algorithm discovers association rules in data. For example, "if a customer purchases a razor and after shave, then that customer will purchase shaving cream with 80% confidence." The association mining problem can be decomposed into two subproblems:

- Find all combinations of items, called frequent itemsets, whose support is greater than the minimum support.
- Use the frequent itemsets to generate the desired rules. The idea is that if, for example, ABC and BC are frequent, then the rule "A implies BC" holds if the ratio of **support (ABC)** to **support (BC)** is at least as large as the minimum confidence. Note that the rule will have minimum support because ABCD is frequent. ODM Association only supports single consequent rules (ABC implies D).

The number of frequent itemsets is governed by the minimum support parameters. The number of rules generated is governed by the number of frequent itemsets and the confidence parameter. If the confidence parameter is set too high, there may be frequent itemsets in the association model but no rules.

ODM uses an SQL-based implementation of the Apriori algorithm. The candidate generation and support counting steps are implemented using SQL queries. Specialized in-memory data structures are not used. The SQL queries are fine-tuned to run efficiently in the database server by using various hints.

- [Apriori Fields Options](#)
- [Apriori Model Options](#)

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle O-Cluster](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [k-Means Model Options](#)
- [k-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)
- [AI Selection Options](#)
- [Oracle Data Mining Examples](#)

Apriori Fields Options

All modeling nodes have a Fields tab, where you can specify the fields to be used in building the model.

Before you can build an Apriori model, you need to specify which fields you want to use as the items of interest in association modeling.

Use type node settings. This option tells the node to use field information from an upstream Type node. This is the default.

Use custom settings. This option tells the node to use field information specified here instead of that given in any upstream Type node(s). After selecting this option, specify the remaining fields on the dialog, which depend on whether you are using transactional format.

If you are *not* using transactional format, specify:

- **Inputs.** Select the input field(s). This is similar to setting a field role to *Input* in a Type node.
- **Partition.** This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building.

If you are using transactional format, specify:

Use transactional format. Use this option if you want to transform data from a row per item, to a row per case.

Selecting this option changes the field controls in the lower part of this dialog box:

For transactional format, specify:

- **ID.** Select an ID field from the list. Numeric or symbolic fields can be used as the ID field. Each unique value of this field should indicate a specific unit of analysis. For example, in a market basket application, each ID might represent a single customer. For a Web log analysis application, each ID might represent a computer (by IP address) or a user (by login data).
- **Content.** Specify the content field for the model. This field contains the item of interest in association modeling.
- **Partition.** This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to create the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)

Apriori Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM® SPSS® Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11g only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Maximum rule length. Sets the maximum number of preconditions for any rule, an integer from 2 to 20. This is a way to limit the complexity of the rules. If the rules are too complex or too specific, or if your rule set is taking too long to train, try decreasing this setting.

Minimum confidence. Sets the minimum confidence level, a value between 0 and 1. Rules with lower confidence than the specified criterion are discarded.

Minimum support. Sets the minimum support threshold, a value between 0 and 1. Apriori discovers patterns with frequency above the minimum support threshold.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Apriori](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle O-Cluster](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)

- [AI Selection Options](#)
-

Oracle Minimum Description Length (MDL)

The Oracle Minimum Description Length (MDL) algorithm helps to identify the attributes that have the greatest influence on a target attribute. Oftentimes, knowing which attributes are most influential helps you to better understand and manage your business and can help simplify modeling activities. Additionally, these attributes can indicate the types of data you may wish to add to augment your models. MDL might be used, for example, to find the process attributes most relevant to predicting the quality of a manufactured part, the factors associated with churn, or the genes most likely involved in the treatment of a particular disease.

Oracle MDL discards input fields that it regards as unimportant in predicting the target. With the remaining input fields it then builds an unrefined model nugget that is associated with an Oracle model, visible in Oracle Data Miner. Browsing the model in Oracle Data Miner displays a chart showing the remaining input fields, ranked in order of their significance in predicting the target.

Negative ranking indicates noise. Input fields ranked at zero or less do not contribute to the prediction and should probably be removed from the data.

To display the chart

1. Right-click on the unrefined model nugget in the Models palette and choose Browse.
2. From the model window, click the button to launch Oracle Data Miner.
3. Connect to Oracle Data Miner. See the topic [Oracle Data Miner](#) for more information.
4. In the Oracle Data Miner navigator panel, expand Models, then Attribute Importance.
5. Select the relevant Oracle model (it will have the same name as the target field you specified in IBM® SPSS® Modeler). If you are not sure which is the correct one, select the Attribute Importance folder and look for a model by creation date.

- [MDL Model Options](#)

Related information

- [Oracle Naïve Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naïve Bayes Model Options](#)
- [Naïve Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [MDL Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)
- [AI Selection Options](#)
- [Oracle Data Mining Examples](#)

MDL Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM® SPSS® Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11g only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Apriori](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle O-Cluster](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [K-Means Model Options](#)
- [K-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)
- [Apriori Model Options](#)
- [Oracle Attribute Importance \(AI\)](#)
- [AI Model Options](#)
- [AI Selection Options](#)

Oracle Attribute Importance (AI)

The objective of attribute importance is to find out which attributes in the data set are related to the result, and the degree to which they influence the final outcome. The Oracle Attribute Importance node analyzes data, finds patterns, and predicts outcomes or results with an associated level of confidence.

- [AI Model Options](#)
- [AI Selection Options](#)
- [AI Model Nugget Model Tab](#)

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)

- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Models Server Options](#)
 - [Misclassification Costs](#)
 - [Naive Bayes Model Options](#)
 - [Naive Bayes Expert Options](#)
 - [Adaptive Bayes Model Options](#)
 - [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [k-Means Model Options](#)
 - [k-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [AI Model Options](#)
 - [AI Selection Options](#)
 - [Oracle Data Mining Examples](#)
-

AI Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Auto Data Preparation. (11g only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Models Server Options](#)
- [Misclassification Costs](#)
- [Naive Bayes Model Options](#)
- [Naive Bayes Expert Options](#)
- [Adaptive Bayes Model Options](#)
- [Adaptive Bayes Expert Options](#)
- [Oracle SVM Model Options](#)
- [Oracle SVM Expert Options](#)
- [Oracle SVM Weights Options](#)
- [Oracle Generalized Linear Models \(GLM\)](#)
- [Decision Tree Model Options](#)
- [Decision Tree Expert Options](#)
- [O-Cluster Model Options](#)
- [O-Cluster Expert Options](#)
- [k-Means Model Options](#)
- [k-Means Expert Options](#)
- [NMF Model Options](#)
- [NMF Expert Options](#)

- [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Selection Options](#)
 - [Oracle Data Mining Examples](#)
-

AI Selection Options

The Options tab allows you to specify the default settings for selecting or excluding input fields in the model nugget. You can then add the model to a stream to select a subset of fields for use in subsequent model-building efforts. Alternatively, you can override these settings by selecting or deselecting additional fields in the model browser after generating the model. However, the default settings make it possible to apply the model nugget without further changes, which may be particularly useful for scripting purposes.

The following options are available:

All fields ranked. Selects fields based on their ranking as *important*, *marginal*, or *unimportant*. You can edit the label for each ranking as well as the cutoff values used to assign records to one rank or another.

Top number of fields. Selects the top *n* fields based on importance.

Importance greater than. Selects all fields with importance greater than the specified value.

The target field is always preserved regardless of the selection.

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Decision Tree](#)
 - [Oracle O-Cluster](#)
 - [Oracle k-Means](#)
 - [Oracle Apriori](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle Models Server Options](#)
 - [Misclassification Costs](#)
 - [Naive Bayes Model Options](#)
 - [Naive Bayes Expert Options](#)
 - [Adaptive Bayes Model Options](#)
 - [Adaptive Bayes Expert Options](#)
 - [Oracle SVM Model Options](#)
 - [Oracle SVM Expert Options](#)
 - [Oracle SVM Weights Options](#)
 - [Oracle Generalized Linear Models \(GLM\)](#)
 - [Decision Tree Model Options](#)
 - [Decision Tree Expert Options](#)
 - [O-Cluster Model Options](#)
 - [O-Cluster Expert Options](#)
 - [K-Means Model Options](#)
 - [K-Means Expert Options](#)
 - [NMF Model Options](#)
 - [NMF Expert Options](#)
 - [Apriori Model Options](#)
 - [MDL Model Options](#)
 - [Oracle Attribute Importance \(AI\)](#)
 - [AI Model Options](#)
 - [Oracle Data Mining Examples](#)
-

AI Model Nugget Model Tab

The Model tab for an Oracle AI model nugget displays the rank and importance of all inputs, and allows you to select fields for filtering by using the check boxes in the column on the left. When you run the stream, only the checked fields are preserved, together with the target prediction.

The other input fields are discarded. The default selections are based on the options specified in the modeling node, but you can select or deselect additional fields as needed.

- To sort the list by rank, field name, importance, or any of the other displayed columns, click on the column header. Alternatively, select the desired item from the list next to the Sort By button, and use the up and down arrows to change the direction of the sort.
- You can use the toolbar to check or uncheck all fields and to access the Check Fields dialog box, which allows you to select fields by rank or importance. You can also press the Shift or Ctrl keys while clicking on fields to extend the selection.
- The threshold values for ranking inputs as important, marginal, or unimportant are displayed in the legend below the table. These values are specified in the modeling node.

Related information

- [Oracle Naïve Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Decision Tree](#)
- [Oracle O-Cluster](#)
- [Oracle k-Means](#)
- [Oracle Apriori](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle Data Mining Examples](#)

Managing Oracle Models

Oracle models are added to the Models palette just like other IBM® SPSS® Modeler models and can be used in much the same way. However, there are a few important differences, given that each Oracle model created in IBM SPSS Modeler actually references a model stored on a database server.

- [Oracle Model Nugget Server Tab](#)
- [Oracle Model Nugget Summary Tab](#)
- [Oracle Model Nugget Settings Tab](#)
- [Listing Oracle Models](#)
- [Oracle Data Miner](#)

Related information

- [About Oracle Data Mining](#)
- [Oracle Model Nugget Server Tab](#)
- [Oracle Model Nugget Summary Tab](#)
- [Oracle Model Nugget Settings Tab](#)
- [Listing Oracle Models](#)
- [Preparing the Data](#)
- [Oracle Naïve Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Apriori](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle O-Cluster](#)

Oracle Model Nugget Server Tab

Building an ODM model via IBM® SPSS® Modeler creates a model in IBM SPSS Modeler and creates or replaces a model in the Oracle database. An IBM SPSS Modeler model of this kind references the content of a database model stored on a database server. IBM SPSS Modeler can perform consistency checking by storing an identical generated **model key** string in both the IBM SPSS Modeler model and the Oracle model.

The key string for each Oracle model is displayed under the *Model Information* column in the List Models dialog box. The key string for an IBM SPSS Modeler model is displayed as the Model Key on the Server tab of an IBM SPSS Modeler model (when placed into a stream).

The Check button on the Server tab of a model nugget can be used to check that the model keys in the IBM SPSS Modeler model and the Oracle model match. If no model of the same name can be found in Oracle or if the model keys do not match, the Oracle model has been deleted or rebuilt since the IBM SPSS Modeler model was built.

Related information

- [About Oracle Data Mining](#)
 - [Managing Oracle Models](#)
 - [Oracle Model Nugget Summary Tab](#)
 - [Oracle Model Nugget Settings Tab](#)
 - [Listing Oracle Models](#)
 - [Preparing the Data](#)
 - [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Apriori](#)
 - [Oracle Decision Tree](#)
 - [Oracle k-Means](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle O-Cluster](#)
-

Oracle Model Nugget Summary Tab

The Summary tab of a model nugget displays information about the model itself (*Analysis*), fields used in the model (*Fields*), settings used when building the model (*Build Settings*), and model training (*Training Summary*).

When you first browse the node, the Summary tab results are collapsed. To see the results of interest, use the expander control to the left of an item to unfold it or click the Expand All button to show all results. To hide the results when you have finished viewing them, use the expander control to collapse the specific results you want to hide or click the Collapse All button to collapse all results.

Analysis. Displays information about the specific model. If you have executed an Analysis node attached to this model nugget, information from that analysis will also appear in this section.

Fields. Lists the fields used as the target and the inputs in building the model.

Build Settings. Contains information about the settings used in building the model.

Training Summary. Shows the type of model, the stream used to create it, the user who created it, when it was built, and the elapsed time for building the model.

Related information

- [About Oracle Data Mining](#)
 - [Managing Oracle Models](#)
 - [Oracle Model Nugget Server Tab](#)
 - [Oracle Model Nugget Settings Tab](#)
 - [Listing Oracle Models](#)
 - [Preparing the Data](#)
 - [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Apriori](#)
 - [Oracle Decision Tree](#)
 - [Oracle k-Means](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle O-Cluster](#)
-

Oracle Model Nugget Settings Tab

The Settings tab on the model nugget allows you to override the setting of certain options on the modeling node for scoring purposes.

Oracle Decision Tree

Use misclassification costs. Determines whether to use misclassification costs in the Oracle Decision Tree model. See the topic [Misclassification Costs](#) for more information.

Rule identifier. If selected (checked), adds a rule identifier column to the Oracle Decision Tree model. The rule identifier identifies the node in the tree at which a particular split is made.

Oracle NMF

Display all features. If selected (checked), displays the feature ID and confidence for all features, instead of those values for only the best feature, in the Oracle NMF model.

Related information

- [Oracle Decision Tree](#)
 - [Managing Oracle Models](#)
 - [Oracle Model Nugget Server Tab](#)
 - [Oracle Model Nugget Summary Tab](#)
 - [Listing Oracle Models](#)
 - [About Oracle Data Mining](#)
 - [Preparing the Data](#)
-

Listing Oracle Models

The List Oracle Data Mining Models button launches a dialog box that lists the existing database models and allows models to be removed. This dialog box can be launched from the Helper Applications dialog box and from the build, browse, and apply dialog boxes for ODM-related nodes.

The following information is displayed for each model:

- **Model Name.** Name of the model, which is used to sort the list
- **Model Information.** Model key information composed of build date/time and target column name
- **Model Type.** Name of the algorithm that built this model

Related information

- [About Oracle Data Mining](#)
 - [Managing Oracle Models](#)
 - [Oracle Model Nugget Server Tab](#)
 - [Oracle Model Nugget Summary Tab](#)
 - [Oracle Model Nugget Settings Tab](#)
 - [Preparing the Data](#)
 - [Oracle Naïve Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Apriori](#)
 - [Oracle Decision Tree](#)
 - [Oracle k-Means](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle O-Cluster](#)
-

Oracle Data Miner

Oracle Data Miner is the user interface to Oracle Data Mining (ODM) and replaces the previous IBM® SPSS® Modeler user interface to ODM. Oracle Data Miner is designed to increase the analyst's success rate in properly utilizing ODM algorithms. These goals are addressed in several ways:

- Users need more assistance in applying a methodology that addresses both data preparation and algorithm selection. Oracle Data Miner meets this need by providing Data Mining Activities to step users through the proper methodology.
- Oracle Data Miner includes improved and expanded heuristics in the model building and transformation wizards to reduce the chance of error in specifying model and transformation settings.

Defining an Oracle Data Miner Connection

1. Oracle Data Miner can be launched from all Oracle build, apply nodes, and output dialog boxes via the **Launch Oracle Data Miner** button.

Figure 1. Launch Oracle Data Miner Button



2. The Oracle Data Miner **Edit Connection** dialog box is presented to the user before the Oracle Data Miner external application is launched (provided the Helper Application option is properly defined).

Note: This dialog box only displays in the absence of a defined connection name.

- Provide a Data Miner connection name and enter the appropriate Oracle 10gR1 or 10gR2 server information. The Oracle server should be the same server specified in IBM SPSS Modeler.

3. The Oracle Data Miner **Choose Connection** dialog box provides options for specifying which connection name, defined in the above step, is used.

Refer to [Oracle Data Miner](#) on the Oracle Web site for more information regarding Oracle Data Miner requirements, installation, and usage.

Preparing the Data

Two types of data preparation may be useful when you are using the Naive Bayes, Adaptive Bayes, and Support Vector Machine provided with Oracle Data Mining algorithms in modeling:

- **Binning**, or conversion of continuous numeric range fields to categories for algorithms that cannot accept continuous data.
- **Normalization**, or transformations applied to numeric ranges so that they have similar means and standard deviations.

Binning

IBM® SPSS® Modeler's Binning node offers a number of techniques for performing binning operations. A binning operation is defined that can be applied to one or many fields. Executing the binning operation on a dataset creates the thresholds and allows an IBM SPSS Modeler Derive node to be created. The derive operation can be converted to SQL and applied prior to model building and scoring. This approach creates a dependency between the model and the Derive node that performs the binning but allows the binning specifications to be reused by multiple modeling tasks.

Normalization

Continuous (numeric range) fields that are used as inputs to Support Vector Machine models should be normalized prior to model building. In the case of regression models, normalization must also be reversed to reconstruct the score from the model output. The SVM model settings allow you to choose Z-Score, Min-Max, or None. The normalization coefficients are constructed by Oracle as a step in the model-building process, and the coefficients are uploaded to IBM SPSS Modeler and stored with the model. At apply time, the coefficients are converted into IBM SPSS Modeler derive expressions and used to prepare the data for scoring before passing the data to the model. In this case, normalization is closely associated with the modeling task.

Related information

- [About Oracle Data Mining](#)
- [Managing Oracle Models](#)
- [Oracle Model Nugget Server Tab](#)
- [Oracle Model Nugget Summary Tab](#)
- [Oracle Model Nugget Settings Tab](#)
- [Listing Oracle Models](#)
- [Oracle Naïve Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Apriori](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle O-Cluster](#)

Oracle Data Mining Examples

A number of sample streams are included that demonstrate the use of ODM with IBM® SPSS® Modeler. These streams can be found in the IBM SPSS Modeler installation folder under `|Demos|Database_Modelling|Oracle Data Mining|`.

Note: The Demos folder can be accessed from the IBM SPSS Modeler program group on the Windows Start menu.

The streams in the following table can be used together in sequence as an example of the database mining process, using the Support Vector Machine (SVM) algorithm that is provided with Oracle Data Mining:

Table 1. Database mining - example streams

Stream	Description
<code>1_upload_data.str</code>	Used to clean and upload data from a flat file into the database.
<code>2_explore_data.str</code>	Provides an example of data exploration with IBM SPSS Modeler
<code>3_build_model.str</code>	Builds the model using the database-native algorithm.
<code>4_evaluate_model.str</code>	Used as an example of model evaluation with IBM SPSS Modeler
<code>5_deploy_model.str</code>	Deploys the model for in-database scoring.

Note: In order to run the example, streams must be executed in order. In addition, source and modeling nodes in each stream must be updated to reference a valid data source for the database you want to use.

The dataset used in the example streams concerns credit card applications and presents a classification problem with a mixture of categorical and continuous predictors. For more information about this dataset, see the `crx.names` file in the same folder as the sample streams.

This dataset is available from the UCI Machine Learning Repository at <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/credit-screening/>.

- [Example Stream: Upload Data](#)
- [Example Stream: Explore Data](#)
- [Example Stream: Build Model](#)
- [Example Stream: Evaluate Model](#)
- [Example Stream: Deploy Model](#)

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Apriori](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle O-Cluster](#)
- [Example Stream: Upload Data](#)
- [Example Stream: Explore Data](#)
- [Example Stream: Build Model](#)
- [Example Stream: Evaluate Model](#)
- [Example Stream: Deploy Model](#)

Example Stream: Upload Data

The first example stream, `1_upload_data.str`, is used to clean and upload data from a flat file into Oracle.

Since Oracle Data Mining requires a unique ID field, this initial stream uses a Derive node to add a new field to the dataset called `ID`, with unique values 1,2,3, using the IBM® SPSS® Modeler @INDEX function.

The Filler node is used for missing-value handling and replaces empty fields that are read from the text file `crx.data` with `NULL` values.

Related information

- [Oracle Naive Bayes](#)
- [Oracle Adaptive Bayes](#)
- [Oracle Support Vector Machine \(SVM\)](#)
- [Oracle Apriori](#)
- [Oracle Decision Tree](#)
- [Oracle k-Means](#)
- [Oracle Minimum Description Length \(MDL\)](#)
- [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
- [Oracle O-Cluster](#)
- [Oracle Data Mining Examples](#)
- [Example Stream: Explore Data](#)

- [Example Stream: Build Model](#)
 - [Example Stream: Evaluate Model](#)
 - [Example Stream: Deploy Model](#)
-

Example Stream: Explore Data

The second example stream, *2_explore_data.str*, is used to demonstrate use of a Data Audit node to gain a general overview of the data, including summary statistics and graphs.

Double-clicking a graph in the Data Audit Report produces a more detailed graph for deeper exploration of a given field.

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Apriori](#)
 - [Oracle Decision Tree](#)
 - [Oracle k-Means](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle O-Cluster](#)
 - [Oracle Data Mining Examples](#)
 - [Example Stream: Upload Data](#)
 - [Example Stream: Build Model](#)
 - [Example Stream: Evaluate Model](#)
 - [Example Stream: Deploy Model](#)
-

Example Stream: Build Model

The third example stream, *3_build_model.str*, illustrates model building in IBM® SPSS® Modeler. Double-click the Database source node (labeled CREDIT) to specify the data source. To specify build settings, double-click the build node (initially labeled CLASS, which changes to FIELD16 when the data source is specified).

On the Model tab of the dialog box:

1. Ensure that ID is selected as the Unique field.
2. Ensure that Linear is selected as the kernel function and Z-Score is the normalization method.

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Apriori](#)
 - [Oracle Decision Tree](#)
 - [Oracle k-Means](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle O-Cluster](#)
 - [Oracle Data Mining Examples](#)
 - [Example Stream: Upload Data](#)
 - [Example Stream: Explore Data](#)
 - [Example Stream: Evaluate Model](#)
 - [Example Stream: Deploy Model](#)
-

Example Stream: Evaluate Model

The fourth example stream, *4_evaluate_model.str*, illustrates the advantages of using IBM® SPSS® Modeler for in-database modeling. Once you have executed the model, you can add it back to your data stream and evaluate the model by using several tools offered in IBM SPSS Modeler.

Viewing Modeling Results

Attach a Table node to the model nugget to explore your results. The \$O-field16 field shows the predicted value for *field16* in each case, and the \$OC-field16 field shows the confidence value for this prediction.

Evaluating Model Results

You can use the Analysis node to create a coincidence matrix showing the pattern of matches between each predicted field and its target field. Run the Analysis node to see the results.

You can use the Evaluation node to create a gains chart designed to show accuracy improvements made by the model. Run the Evaluation node to see the results.

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Apriori](#)
 - [Oracle Decision Tree](#)
 - [Oracle k-Means](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle O-Cluster](#)
 - [Oracle Data Mining Examples](#)
 - [Example Stream: Upload Data](#)
 - [Example Stream: Explore Data](#)
 - [Example Stream: Build Model](#)
 - [Example Stream: Deploy Model](#)
-

Example Stream: Deploy Model

Once you are satisfied with the accuracy of the model, you can deploy it for use with external applications or for publishing back to the database. In the final example stream, *5_deploy_model.str*, data are read from the table **CREDITDATA** and then scored and published to the table **CREDITSCORES** using the Publisher node called *deploy solution*.

Related information

- [Oracle Naive Bayes](#)
 - [Oracle Adaptive Bayes](#)
 - [Oracle Support Vector Machine \(SVM\)](#)
 - [Oracle Apriori](#)
 - [Oracle Decision Tree](#)
 - [Oracle k-Means](#)
 - [Oracle Minimum Description Length \(MDL\)](#)
 - [Oracle Nonnegative Matrix Factorization \(NMF\)](#)
 - [Oracle O-Cluster](#)
 - [Oracle Data Mining Examples](#)
 - [Example Stream: Upload Data](#)
 - [Example Stream: Explore Data](#)
 - [Example Stream: Build Model](#)
 - [Example Stream: Evaluate Model](#)
-

Database Modeling with IBM® Netezza® and IBM Netezza Analytics

- [SPSS Modeler with IBM Data Warehouse and IBM Netezza Analytics](#)
- [Integration requirements](#)
- [Enabling integration](#)
- [Building models with IBM Netezza Analytics and IBM Data Warehouse](#)
- [IBM Data WH Regression Tree](#)
- [Netezza Divisive Clustering](#)
- [IBM Data WH Generalized Linear](#)
- [IBM Data WH Decision Trees](#)

- [IBM Data WH Linear Regression](#)
 - [IBM Data WH KNN](#)
 - [IBM Data WH K-Means](#)
 - [IBM Data WH Naive Bayes](#)
 - [Netezza Bayes Net](#)
 - [Netezza Time Series](#)
 - [IBM Data WH TwoStep](#)
 - [IBM Data WH PCA](#)
 - [Managing IBM Data WH and Netezza Models](#)
-

SPSS Modeler with IBM Data Warehouse and IBM Netezza Analytics

IBM® SPSS® Modeler supports integration with IBM Data Warehouse and IBM Netezza® Analytics, which provides the ability to run advanced analytics on those IBM servers. These features can be accessed through the IBM SPSS Modeler graphical user interface and workflow-oriented development environment, allowing you to run the data mining algorithms directly in the IBM Netezza or IBM Data Warehouse environment.

SPSS Modeler supports integration of the following algorithms from **IBM Netezza Analytics**:

- Decision Trees
- K-Means
- TwoStep
- Bayes Net
- Naive Bayes
- KNN
- Divisive Clustering
- PCA
- Regression Tree
- Linear Regression
- Time Series
- Generalized Linear

For more information about these algorithms, see the *IBM Netezza Analytics Developer's Guide* and the *IBM Netezza Analytics Reference Guide*.

SPSS Modeler supports integration of the following algorithms from **IBM Data Warehouse** (Bayes Net, Divisive Clustering, and Time Series are not supported):

- Decision Trees
- K-Means
- TwoStep
- Naive Bayes
- KNN
- PCA
- Regression Tree
- Linear Regression
- Generalized Linear

Note: AIX isn't supported.

Integration requirements

The following conditions are prerequisites for conducting in-database modeling using IBM® Netezza® Analytics or IBM Data Warehouse. You may need to consult with your database administrator to ensure that these conditions are met.

- IBM SPSS® Modeler running against an IBM SPSS Modeler Server installation on Windows or UNIX (except zLinux, for which IBM Netezza ODBC drivers are not available).
- IBM Netezza Performance Server, running the IBM Netezza Analytics package.

Note: The minimum version of Netezza Performance Server (NPS) that is required depends on the version of INZA that is required and is as follows:

- Anything greater than NPS 6.0.0 P8 will support INZA versions prior to 2.0.
- To use INZA 2.0 or greater requires NPS 6.0.5 P5 or greater.

Netezza Generalized Linear and Netezza Time Series require INZA 2.0 and above to be functional. All the other Netezza In-Database nodes need INZA 1.1 or later.

- An ODBC data source for connecting to an IBM Netezza database. See the topic [Enabling integration](#) for more information.
- An ODBC data source for connecting to an IBM Data Warehouse database.

- SQL generation and optimization enabled in IBM SPSS Modeler. See the topic [Enabling integration](#) for more information.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.
Help>About>Additional Details

If connectivity is enabled, you see the option Server Enablement in the License Status tab.

See [Connecting to IBM SPSS Modeler Server](#) for more information.

Enabling integration

Enabling integration with IBM® Netezza® Analytics or IBM Data Warehouse consists of the following steps.

- Configuring IBM Netezza Analytics or IBM Data Warehouse
- Creating an ODBC source
- Enabling the integration in IBM SPSS® Modeler
- Enabling SQL generation and optimization in IBM SPSS Modeler

These are described in the sections that follow.

- [Configuring IBM Netezza Analytics or IBM Data Warehouse](#)
- [Creating an ODBC Source for IBM Netezza Analytics](#)
- [Enabling integration in SPSS Modeler](#)
- [Enabling SQL Generation and Optimization](#)

Configuring IBM Netezza Analytics or IBM Data Warehouse

To install and configure IBM® Netezza® Analytics or IBM Data Warehouse, refer to the appropriate IBM documentation. For example, for IBM Netezza Analytics, see the *IBM Netezza Analytics Installation Guide* provided with that product. The section *Setting Database Permissions* in that guide contains details of scripts that need to be run to allow IBM SPSS® Modeler streams to write to the database.

Note: If you will be using nodes that rely on matrix calculation, the Matrix Engine must be initialized by running `CALL NZM..INITIALIZE()`; otherwise execution of stored procedures will fail. Initialization is a one-time setup step for each database.

Creating an ODBC Source for IBM Netezza® Analytics

To enable the connection between the IBM Netezza database and IBM® SPSS® Modeler, you need to create an ODBC system data source name (DSN).

Before creating a DSN, you should have a basic understanding of ODBC data sources and drivers, and database support in IBM SPSS Modeler.

If you are running in distributed mode against IBM SPSS Modeler Server, create the DSN on the server computer. If you are running in local (client) mode, create the DSN on the client computer.

Windows clients

1. From your *Netezza Client* CD, run the *nzodbcsetup.exe* file to start the installer. Follow the on-screen instructions to install the driver. For full instructions, see the *IBM Netezza ODBC, JDBC, and OLE DB Installation and Configuration Guide*.
 - a. Create the DSN.
Note: The menu sequence depends on your version of Windows.
 - Windows XP. From the Start menu, choose Control Panel. Double-click Administrative Tools, and then double-click Data Sources (ODBC).
 - Windows Vista. From the Start menu, choose Control Panel, then System Maintenance. Double-click Administrative Tools, selectData Sources (ODBC), then click Open.
 - Windows 7. From the Start menu, choose Control Panel, then System & Security, then Administrative Tools. SelectData Sources (ODBC), then click Open.
 - b. Go to the System DSN tab, and then click Add.
2. Select *NetezzaSQL* from the list and click Finish.
3. On the DSN Options tab of the Netezza ODBC Driver Setup screen, type a data source name of your choosing, the hostname or IP address of the IBM Netezza server, the port number for the connection, the database of the IBM Netezza instance you are using, and your

- username and password details for the database connection. Click the Help button for an explanation of the fields.
4. Click the Test Connection button and ensure that you can connect to the database.
 5. When you have a successful connection, click OK repeatedly to exit from the ODBC Data Source Administrator screen.

Windows servers

The procedure for Windows Server is the same as the client procedure for Windows XP.

UNIX or Linux servers

The following procedure applies to UNIX or Linux servers (except zLinux, for which IBM Netezza ODBC drivers are not available).

1. From your Netezza Client CD/DVD, copy the relevant <platform>.cli.package.tar.gz file to a temporary location on the server.
2. Extract the archive contents by means of the **gunzip** and **tar** commands.
3. Add execute permissions to the *unpack* script that is extracted.
4. Run the script, answering the on-screen prompts.
5. Edit the modelersrv.sh file to include the following lines.

```
. <SDAP Install Path>/odbc.sh  
LD_LIBRARY_PATH_64=$LD_LIBRARY_PATH:/usr/local/nz/lib64; export LD_LIBRARY_PATH_64  
NZ_ODBC_INI_PATH=<SDAP Install Path>; export NZ_ODBC_INI_PATH
```

For example:

```
. /usr/IBM/SPSS/SDAP/odbc.sh  
LD_LIBRARY_PATH_64=$LD_LIBRARY_PATH:/usr/local/nz/lib64; export LD_LIBRARY_PATH_64  
NZ_ODBC_INI_PATH=/usr/IBM/SPSS/SDAP; export NZ_ODBC_INI_PATH
```

6. Locate the file /usr/local/nz/lib64/odbc.ini and copy its contents into the odbc.ini file that is installed with SDAP (the one defined by the \$ODBCINI environment variable).

Note: For 64-bit Linux systems, the Driver parameter incorrectly references the 32-bit driver. When you copy the odbc.ini contents in the previous step, edit the path within this parameter accordingly, for example:

```
/usr/local/nz/lib64/libnzodbc.so
```

7. Edit the parameters in the Netezza DSN definition to reflect the database to be used.
8. Restart IBM SPSS Modeler Server and test the use of the Netezza in-database mining nodes on the client.

Enabling integration in SPSS Modeler

1. From the IBM® SPSS® Modeler main menu, choose Tools->Options->Helper Applications.
2. Click the IBM Data Warehouse tab.

Enable IBM Data Warehouse Analytics Integration. Enables the Database Modeling palette (if not already displayed) at the bottom of the IBM SPSS Modeler window and adds the nodes for IBM Data Warehouse and Netezza Data Mining algorithms.

IBM Data Warehouse Connection. Click the Edit button and choose the IBM Data Warehouse connection string that you set up when creating the ODBC source. For more information, see the IBM Data Warehouse admin console.

Enabling SQL Generation and Optimization

Because of the likelihood of working with very large data sets, for performance reasons you should enable the SQL generation and optimization options in IBM® SPSS® Modeler.

1. From the IBM SPSS Modeler menus choose: Tools->Stream Properties->Options
2. Click the Optimization option in the navigation pane.
3. Confirm that the Generate SQL option is enabled. This setting is required for database modeling to function.
4. Select Optimize SQL Generation and Optimize other execution (not strictly required but strongly recommended for optimized performance).

Building models with IBM Netezza® Analytics and IBM Data Warehouse

Each of the supported algorithms has a corresponding modeling node. You can access the IBM Data Warehouse and IBM Netezza modeling nodes from the Database Modeling tab on the nodes palette.

Data considerations

Fields in the data source can contain variables of various data types, depending on the modeling node. In IBM® SPSS® Modeler, data types are known as *measurement levels*. The Fields tab of the modeling node uses icons to indicate the permitted measurement level types for its input and target fields.

Target field The target field is the field whose value you are trying to predict. Where a target can be specified, only one of the source data fields can be selected as the target field.

Record ID field Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. If the source data does not include an ID field, you can create this field by means of a Derive node, as the following procedure shows.

1. Select the source node.
2. From the Field Ops tab on the nodes palette, double-click the Derive node.
3. Open the Derive node by double-clicking its icon on the canvas.
4. In the Derive field field, type (for example) **ID**.
5. In the Formula field, type **@INDEX** and click OK.
6. Connect the Derive node to the rest of the stream.

Note: If you retrieve long numeric data from a Netezza database by using the **NUMERIC(18, 0)** data type, SPSS Modeler can sometimes round up the data during import. To avoid this issue, store your data using either the **BIGINT**, or **NUMERIC(36, 0)** data type.

Note: Due to the limitations on the types of fields that can be used, a field with a typeless Measurement level and a role of Record ID does not appear in a Netezza In-Database modeling node (for example, K-Means).

Handling null values

If the input data contains null values, use of some Netezza nodes may result in error messages or long-running streams, so we recommend removing records containing null values. Use the following method.

1. Attach a Select node to the source node.
2. Set the Mode option of the Select node to Discard.
3. Enter the following in the Condition field:

```
@NULL(field1) [or @NULL(field2) [...] or @NULL(fieldN)]
```

Be sure to include every input field.

4. Connect the Select node to the rest of the stream.

Model output

It is possible for a stream containing a Data Warehouse or Netezza modeling node to produce slightly different results each time it is run. This is because the order in which the node reads the source data is not always the same, as the data is read into temporary tables before model building. However, the differences produced by this effect are negligible.

General comments

- In IBM SPSS Collaboration and Deployment Services, it is not possible to create scoring configurations using streams containing IBM Data Warehouse or IBM Netezza database modeling nodes.
- PMML export or import is not possible for models created by the Data Warehouse or Netezza nodes.
- [Field options](#)
- [Server options](#)
- [Model options](#)
- [Managing models](#)
- [Listing database models](#)

Field options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the All button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Target. Choose one field as the target for the prediction. For Generalized Linear models, see also the Trials field on this screen.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

Server options

On the Server tab, you specify the IBM Data Warehouse database where the model is to be built.

IBM Data Warehouse Server Details. Here you specify the connection details for the database you want to use for the model.

- Use upstream connection. (default) Uses the connection details specified in an upstream node, for example the Database source node. This option works only if all upstream nodes are able to use SQL pushback. In this case there is no need to move the data out of the database, as the SQL fully implements all of the upstream nodes.
- Move data to connection. Moves the data to the database you specify here. Doing so allows modeling to work if the data is in another IBM Data Warehouse database, or a database from another vendor, or even if the data is in a flat file. In addition, data is moved back to the database specified here if the data has been extracted because a node did not perform SQL pushback. Click the Edit button to browse for and select a connection.

CAUTION:

IBM® Netezza® Analytics and IBM Data Warehouse is typically used with very large data sets. Transferring large amounts of data between databases, or out of and back into a database, can be very time-consuming and should be avoided where possible.

Note: The ODBC data source name is effectively embedded in each IBM SPSS® Modeler stream. If a stream that is created on one host is executed on a different host, the name of the data source must be the same on each host. Alternatively, a different data source can be selected on the Server tab in each source or modeling node.

Model options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also set default values for scoring options.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Replace existing if the name has been used. If you select this check box, any existing model of the same name will be overwritten.

Make Available for Scoring. You can set the default values here for the scoring options that appear on the dialog for the model nugget. For details of the options, see the help topic for the Settings tab of that particular nugget.

Managing models

Building an IBM Netezza or IBM Data Warehouse model via SPSS® Modeler creates a model in SPSS Modeler and creates or replaces a model in the IBM Data Warehouse database. The SPSS Modeler model of this kind references the content of a database model stored on a database server. SPSS Modeler can perform consistency checking by storing an identical generated model key string in both the SPSS Modeler model and the Netezza or Data Warehouse model.

The model name for each Netezza or Data Warehouse model is displayed under the *Model Information* column in the Listing Database Models dialog box. The model name for an SPSS Modeler model is displayed as the Model Key on the Server tab of an SPSS Modeler model (when placed into a stream).

The Check button can be used to check that the model keys in the SPSS Modeler model and the Netezza or Data Warehouse model match. If no model of the same name can be found in Netezza or Data Warehouse, or if the model keys do not match, the Netezza or Data Warehouse model has been deleted or rebuilt since the SPSS Modeler model was built.

Listing database models

SPSS® Modeler provides a dialog box for listing the models that are stored in IBM Data Warehouse and enables models to be deleted. This dialog box is accessible from the IBM Helper Applications dialog box and from the build, browse, and apply dialog boxes for IBM Data Warehouse and IBM Netezza data mining-related nodes. The following information is displayed for each model:

- Model name (name of the model, which is used to sort the list).
 - Owner name.
 - The algorithm used in the model.
 - The current state of the model; for example, Complete.
 - The date on which the model was created.
-

IBM Data WH Regression Tree

A regression tree is a tree-based algorithm that splits a sample of cases repeatedly to derive subsets of the same kind, based on values of a numeric target field. As with decision trees, regression trees decompose the data into subsets in which the leaves of the tree correspond to sufficiently small or sufficiently uniform subsets. Splits are selected to decrease the dispersion of target attribute values, so that they can be reasonably well predicted by their mean values at leaves.

- [IBM Data WH Regression Tree Build Options - Tree Growth](#)
 - [IBM Data WH Tree Build Options - Tree Pruning](#)
-

IBM Data WH Regression Tree Build Options - Tree Growth

You can set build options for tree growth and tree pruning.

The following build options are available for tree growth:

Maximum tree depth. The maximum number of levels to which the tree can grow below the root node, that is, the number of times the sample is split recursively. The default is 62, which is the maximum tree depth for modeling purposes.

Note: If the viewer in the model nugget shows the textual representation of the model, a maximum of 12 levels of the tree is displayed.

Splitting Criteria. These options control when to stop splitting the tree. If you do not want to use the default values, click Customize and change the values.

- **Split evaluation measure.** This class evaluation measure evaluates the best place to split the tree.
Note: Currently, variance is the only possible option.
- **Minimum improvement for splits.** The minimum amount by which impurity must be reduced before a new split is created in the tree. The goal of tree building is to create subgroups with similar output values to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the amount that is specified by the splitting criteria, the branch is not split.
- **Minimum number of instances for a split.** The minimum number of records that can be split. When fewer than this number of unsplittable records remain, no further splits are made. You can use this field to prevent the creation of small subgroups in the tree.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- **All.** All column-related statistics and all value-related statistics are included.
Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify None.
 - **Columns.** Column-related statistics are included.
 - **None.** Only statistics that are required to score the model are included.
-

IBM Data WH Tree Build Options - Tree Pruning

You can use the pruning options to specify pruning criteria for the regression tree. The intention of pruning is to reduce the risk of overfitting by removing overgrown subgroups that do not improve the expected accuracy on new data.

Pruning measure. The pruning measure ensures that the estimated accuracy of the model remains within acceptable limits after removing a leaf from the tree. You can select one of the following measures.

- **mse.** Mean squared error - (default) measures how close a fitted line is to the data points.
- **r2.** R-squared - measures the proportion of variation in the dependent variable explained by the regression model.
- **Pearson.** Pearson's correlation coefficient - measures the strength of relationship between linearly dependent variables that are normally distributed.
- **Spearman.** Spearman's correlation coefficient - detects nonlinear relationships that appear weak according to Pearson's correlation, but which may actually be strong.

Data for pruning. You can use some or all of the training data to estimate the expected accuracy on new data. Alternatively, you can use a separate pruning dataset from a specified table for this purpose.

- Use all training data. This option (the default) uses all the training data to estimate the model accuracy.
- Use % of training data for pruning. Use this option to split the data into two sets, one for training and one for pruning, using the percentage specified here for the pruning data.
Select Replicate results if you want to specify a random seed to ensure that the data is partitioned in the same way each time you run the stream. You can either specify an integer in the Seed used for pruning field, or click Generate, which will create a pseudo-random integer.
- Use data from an existing table. Specify the table name of a separate pruning dataset for estimating model accuracy. Doing so is considered more reliable than using training data. However, this option may result in the removal of a large subset of data from the training set, thus reducing the quality of the decision tree.

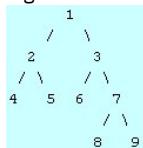
Netezza Divisive Clustering

Divisive clustering is a method of cluster analysis in which the algorithm is run repeatedly to divide clusters into subclusters until a specified stopping point is reached.

Cluster formation begins with a single cluster containing all training instances (records). The first iteration of the algorithm divides the data set into two subclusters, with subsequent iterations dividing these into further subclusters. The stopping criteria are specified as a maximum number of iterations, a maximum number of levels to which the data set is divided, and a minimum required number of instances for further partitioning.

The resulting hierarchical clustering tree can be used to classify instances by propagating them down from the root cluster, as in the following example.

Figure 1. Example of a divisive clustering tree



At each level, the best matching subcluster is chosen with respect to the distance of the instance from the subcluster centers.

When the instances are scored with an applied hierarchy level of -1 (the default), the scoring returns only a leaf cluster, as leaves are designated by a negative number. In the example, this would be one of clusters 4, 5, 6, 8, or 9. However, if the hierarchy level is set to 2, for example, scoring would return one of the clusters at the second level below the root cluster, namely 4, 5, 6, or 7.

- [Netezza Divisive Clustering Field Options](#)
- [Netezza Divisive Clustering Build Options](#)

Related information

- [Netezza Divisive Clustering Model Nuggets](#)

Netezza Divisive Clustering Field Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the All button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

Related information

- [Netezza Divisive Clustering](#)
 - [Netezza Divisive Clustering Model Nuggets](#)
-

Netezza Divisive Clustering Build Options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- Euclidean. (default) The distance between two points is computed by joining them with a straight line.
- Manhattan. The distance between two points is calculated as the sum of the absolute differences between their co-ordinates.
- Canberra. Similar to Manhattan distance, but more sensitive to data points closer to the origin.
- Maximum. The distance between two points is calculated as the greatest of their differences along any coordinate dimension.

Maximum number of iterations. The algorithm operates by performing several iterations of the same process. This option allows you to stop model training after the number of iterations specified.

Maximum depth of cluster trees. The maximum number of levels to which the data set can be subdivided.

Replicate results. Check this box if you want to set a random seed, which will enable you to replicate analyses. You can either specify an integer or click Generate, which creates a pseudo-random integer.

Minimum number of instances for a split. The minimum number of records that can be split. When fewer than this number of unsplit records remain, no further splits will be made. You can use this field to prevent the creation of very small subgroups in the cluster tree.

Related information

- [Netezza Divisive Clustering](#)
 - [Netezza Divisive Clustering Model Nuggets](#)
-

IBM Data WH Generalized Linear

Linear regression is a long-established statistical technique for classifying records based on the values of numeric input fields. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values. Linear models are useful in modeling a wide range of real-world phenomena owing to their simplicity in both training and model application. However, linear models assume a normal distribution in the dependent (target) variable and a linear impact of the independent (predictor) variables on the dependent variable.

There are many situations where a linear regression is useful but the above assumptions do not apply. For example, when modeling consumer choice between a discrete number of products, the dependent variable is likely to have a multinomial distribution. Equally, when modeling income against age, income typically increases as age increases, but the link between the two is unlikely to be as simple as a straight line.

For these situations, a generalized linear model can be used. Generalized linear models expand the linear regression model so that the dependent variable is related to the predictor variables by means of a specified link function, for which there is a choice of suitable functions. Moreover, the model allows for the dependent variable to have a non-normal distribution, such as Poisson.

The algorithm iteratively seeks the best-fitting model, up to a specified number of iterations. In calculating the best fit, the error is represented by the sum of squares of the differences between the predicted and actual value of the dependent variable.

- [IBM Data WH Generalized Linear Model Field Options](#)
- [IBM Data WH Generalized Linear Model Options - General](#)
- [IBM Data WH Generalized Linear Model Options - Interaction](#)

- [IBM Data WH Generalized Linear Model Options - Scoring Options](#)

IBM Data WH Generalized Linear Model Field Options

On the Fields tab, you choose whether you want to use the field role settings that are already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings, such as targets or predictors from an upstream Type node, or the Types tab of an upstream source node.

Use custom field assignments. Choose this option if you want to assign targets, predictors, and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the All button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Target. Choose one field as the target for the prediction.

Record ID. The field that is to be used as the unique record identifier. The values of this field must be unique for each record, for example, customer ID numbers.

Instance Weight. Specify a field to use instance weights. An instance weight is a weight per row of input data. By default, all input records are assumed to have equal relative importance. You can change the importance by assigning individual weights to the input records. The field that you specify must contain a numeric weight for each row of input data.

Predictors (Inputs). Select the input field or fields. This action is similar to setting the field role to *Input* in a Type node.

IBM Data WH Generalized Linear Model Options - General

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also make various settings relating to the model, the link function, the input field interactions (if any), and set default values for scoring options.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Field options. You can specify the roles of the input fields for building the model.

General Settings. These settings relate to the stopping criteria for the algorithm.

- **Maximum number of iterations.** The maximum number of iterations the algorithm will perform; minimum is 1, default is 20.
- **Maximum error (1e).** The maximum error value (in scientific notation) at which the algorithm should stop finding the best fit model. Minimum is 0, default is -3, meaning 1E-3, or 0.001.
- **Insignificant error values threshold (1e).** The value (in scientific notation) below which errors are treated as having a value of zero. Minimum is -1, default is -7, meaning that error values below 1E-7 (or 0.0000001) are counted as insignificant.

Distribution Settings. These settings relate to the distribution of the dependent (target) variable.

- **Distribution of response variable.** The distribution type; one of Bernoulli (default), Gaussian, Poisson, Binomial, Negative binomial, Wald (Inverse Gaussian), and Gamma.
- **Parameters.** (Poisson or binomial distribution only) You must specify one of the following options in the Specify parameter field:
 - To automatically have the parameter estimated from data, select Default.
 - To allow optimization of the distribution quasi-likelihood, select Quasi.
 - To explicitly specify the parameter value, select Explicit.(Binomial distribution only) You must specify the input table column that is to be used as the trials field as required by binomial distribution. This column contains the number of trials for the binomial distribution.
(Negative binomial distribution only) You can use the default of -1 or specify a different parameter value.

Link Function Settings. These settings relate to the link function, which relates the dependent variable to the predictor variables.

- **Link function.** The function to be used; one of Identity, Inverse, Invnegative, Invsquare, Sqrt, Power, Oddspower, Log, Clog, Loglog, Cloglog, Logit (default), Probit, Gaussit, Cauchit, Canbinom, Cangeom, Cannegbinom.
- **Parameters.** (Power or Oddspower link functions only) You can specify a parameter value if the link function is Power or Oddspower. Choose to either specify a value, or use the default of 1.

IBM Data WH Generalized Linear Model Options - Interaction

The Interaction panel contains the options for specifying interactions (that is, multiplicative effects between input fields).

Column Interaction. Select this check box to specify interactions between input fields. Leave the box cleared if there are no interactions.

Enter interactions into the model by selecting one or more fields in the source list and dragging to the interactions list. The type of interaction created depends upon the hotspot onto which you drop the selection.

- **Main.** Dropped fields appear as separate main interactions at the bottom of the interactions list.
- **2-way.** All possible pairs of the dropped fields appear as 2-way interactions at the bottom of the interactions list.
- **3-way.** All possible triplets of the dropped fields appear as 3-way interactions at the bottom of the interactions list.
- *****. The combination of all dropped fields appear as a single interaction at the bottom of the interactions list.

Include Intercept. The intercept is usually included in the model. If you can assume the data passes through the origin, you can exclude the intercept.

Dialog box buttons

The buttons to the right of the display enable you to make changes to the terms used in the model.

Figure 1. Delete button



Delete terms from the model by selecting the terms you want to delete and clicking the delete button.

Figure 2. Reorder buttons



Reorder the terms within the model by selecting the terms you want to reorder and clicking the up or down arrow.

Figure 3. Custom interaction button



- [Add a Custom Term](#)

Add a Custom Term

You can specify custom interactions in the form $n1*x1*x1*x1...$ Select a field from the Fields list, click the right-arrow button to add the field to Custom Term, click By*, select the next field, click the right-arrow button, and so on. When you have built the custom interaction, click Add term to return it to the Interaction panel.

IBM Data WH Generalized Linear Model Options - Scoring Options

Make Available for Scoring. You can set the default values here for the scoring options that appear on the dialog for the model nugget. See the topic [IBM Data WH Generalized Linear Model Nugget - Settings tab](#) for more information.

- **Include input fields.** Select this check box if you want to display the input fields in the model output as well as the predictions.

IBM Data WH Decision Trees

A decision tree is a hierarchical structure that represents a classification model. With a decision tree model, you can develop a classification system to predict or classify future observations from a set of training data. The classification takes the form of a tree structure in which the branches represent split points in the classification. The splits break the data down into subgroups recursively until a stopping point is reached. The tree nodes at the stopping points are known as **leaves**. Each leaf assigns a label, known as a **class label**, to the members of its subgroup, or class.

- [Instance weights and class weights](#)
 - [Netezza Decision Tree Field Options](#)
 - [IBM Data WH Decision Tree Build Options](#)
-

Instance weights and class weights

By default, all input records and classes are assumed to have equal relative importance. You can change this by assigning individual weights to the members of either or both of these items. Doing so might be useful, for example, if the data points in your training data are not realistically distributed among the categories. Weights enable you to bias the model so that you can compensate for those categories that are less well represented in the data. Increasing the weight for a target value should increase the percentage of correct predictions for that category.

In the Decision Tree modeling node, you can specify two types of weights. **Instance weights** assign a weight to each row of input data. The weights are typically specified as 1.0 for most cases, with higher or lower values given only to those cases that are more or less important than the majority, as shown in the following table.

Table 1. Instance weight example

Record ID	Target	Instance Weight
1	drugA	1.1
2	drugB	1.0
3	drugA	1.0
4	drugB	0.3

Class weights assign a weight to each category of the target field, as shown in the following table.

Table 2. Class weight example

Class	Class Weight
drugA	1.0
drugB	1.5

Both types of weights can be used at the same time, in which case they are multiplied together and used as instance weights. Thus if the two previous examples were used together, the algorithm would use the instance weights as shown in the following table.

Table 3. Instance weight calculation example

Record ID	Calculation	Instance Weight
1	1.1*1.0	1.1
2	1.0*1.5	1.5
3	1.0*1.0	1.0
4	0.3*1.5	0.45

Netezza Decision Tree Field Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign targets, predictors and other roles, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

To select all the fields in the list, click the All button, or click an individual measurement level button to select all fields with that measurement level.

Target. Select one field as the target for the prediction.

Record ID. The field that is to be used as the unique record identifier. The values of this field must be unique for each record (for example, customer ID numbers).

Instance Weight. Specifying a field here enables you to use instance weights (a weight per row of input data) instead of, or in addition to, the default, class weights (a weight per category for the target field). The field you specify here must be one that contains a numeric weight for each row of input data. See the topic [Instance weights and class weights](#) for more information.

Predictors (Inputs). Select the input field or fields. This is similar to setting the field role to *Input* in a Type node.

Related information

- [IBM Data WH Decision Trees](#)
 - [IBM Data WH Decision Tree Model Nuggets](#)
-

IBM Data WH Decision Tree Build Options

The following build options are available for tree growth:

Growth Measure. These options control the way tree growth is measured.

- **Impurity Measure.** This measure evaluates the best place to split the tree. It is a measurement of the variability in a subgroup or segment of data. A low impurity measurement indicates a group where most members have similar values for the criterion or target field. The supported measurements are Entropy and Gini. These measurements are based on probabilities of category membership for the branch.
- **Maximum tree depth.** The maximum number of levels to which the tree can grow below the root node, that is, the number of times the sample is split recursively. The default value of this property is 10, and the maximal value that you can set for this property is 62. Note: If the viewer in the model nugget shows the textual representation of the model, a maximum of 12 levels of the tree is displayed.

Splitting Criteria. These options control when to stop splitting the tree.

- **Minimum improvement for splits.** The minimum amount by which impurity must be reduced before a new split is created in the tree. The goal of tree building is to create subgroups with similar output values to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the amount that is specified by the splitting criteria, the branch is not split.
- **Minimum number of instances for a split.** The minimum number of records that can be split. When fewer than this number of unsplittable records remain, no further splits are made. You can use this field to prevent the creation of small subgroups in the tree.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- **All.** All column-related statistics and all value-related statistics are included.
Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify None.
 - **Columns.** Column-related statistics are included.
 - **None.** Only statistics that are required to score the model are included.
 - [IBM Data WH Decision Tree Node - Class Weights](#)
 - [IBM Data WH Decision Tree Node - Tree Pruning](#)
-

IBM Data WH Decision Tree Node - Class Weights

Here you can assign weights to individual classes. The default is to assign a value of 1 to all classes, making them equally weighted. By specifying different numerical weights for different class labels, you instruct the algorithm to weight the training sets of particular classes accordingly.

To change a weight, double-click it in the Weight column and make the changes you want.

Value. The set of class labels, derived from the possible values of the target field.

Weight. The weighting to be assigned to a particular class. Assigning a higher weight to a class makes the model more sensitive to that class relative to the other classes.

You can use class weights in combination with instance weights. See the topic [Instance weights and class weights](#) for more information.

IBM Data WH Decision Tree Node - Tree Pruning

You can use the pruning options to specify pruning criteria for the decision tree. The intention of pruning is to reduce the risk of overfitting by removing overgrown subgroups that do not improve the expected accuracy on new data.

Pruning measure. The default pruning measure, Accuracy, ensures that the estimated accuracy of the model remains within acceptable limits after removing a leaf from the tree. Use the alternative, Weighted Accuracy, if you want to take the class weights into account while applying pruning.

Data for pruning. You can use some or all of the training data to estimate the expected accuracy on new data. Alternatively, you can use a separate pruning dataset from a specified table for this purpose.

- Use all training data. This option (the default) uses all the training data to estimate the model accuracy.
- Use % of training data for pruning. Use this option to split the data into two sets, one for training and one for pruning, using the percentage specified here for the pruning data.

Select Replicate results if you want to specify a random seed to ensure that the data is partitioned in the same way each time you run the stream. You can either specify an integer in the Seed used for pruning field, or click Generate, which will create a pseudo-random integer.

- Use data from an existing table. Specify the table name of a separate pruning dataset for estimating model accuracy. Doing so is considered more reliable than using training data. However, this option may result in the removal of a large subset of data from the training set, thus reducing the quality of the decision tree.

IBM Data WH Linear Regression

Linear models predict a continuous target based on linear relationships between the target and one or more predictors. While limited to directly modeling linear relationships only, linear regression models are relatively simple and give an easily interpreted mathematical formula for scoring. Linear models are fast, efficient and easy to use, although their applicability is limited compared to those produced by more refined regression algorithms.

- [IBM Data WH Linear Regression Build Options](#)

IBM Data WH Linear Regression Build Options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

Use Singular Value Decomposition to solve equations. Using the Singular Value Decomposition matrix instead of the original matrix has the advantage of being more robust against numerical errors, and can also speed up computation.

Include intercept in the model. Including the intercept increases the overall accuracy of the solution.

Calculate model diagnostics. This option causes a number of diagnostics to be calculated on the model. The results are stored in matrices or tables for later review. The diagnostics include r-squared, residual sum-of-squares, estimation of variance, standard deviation, *p*-value, and *t*-value.

These diagnostics relate to the validity and usefulness of the model. You should run separate diagnostics on the underlying data to ensure that it meets linearity assumptions.

IBM Data WH KNN

Nearest Neighbor Analysis is a method for classifying cases based on their similarity to other cases. In machine learning, it was developed as a way to recognize patterns of data without requiring an exact match to any stored patterns, or cases. Similar cases are near each other and dissimilar cases are distant from each other. Thus, the distance between two cases is a measure of their dissimilarity.

Cases that are near each other are said to be “neighbors.” When a new case (holdout) is presented, its distance from each of the cases in the model is computed. The classifications of the most similar cases – the nearest neighbors – are tallied and the new case is placed into the category that contains the greatest number of nearest neighbors.

You can specify the number of nearest neighbors to examine; this value is called *k*. The pictures show how a new case would be classified using two different values of *k*. When *k* = 5, the new case is placed in category 1 because a majority of the nearest neighbors belong to category 1. However, when *k* = 9, the new case is placed in category 0 because a majority of the nearest neighbors belong to category 0.

Nearest neighbor analysis can also be used to compute values for a continuous target. In this situation, the average or median target value of the nearest neighbors is used to obtain the predicted value for the new case.

- [IBM Data WH KNN Model Options - General](#)
- [IBM Data WH KNN Model Options - Scoring Options](#)

IBM Data WH KNN Model Options - General

On the Model Options - General tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also set options that control how the number of nearest neighbors is calculated, and set options for enhanced performance and accuracy of the model.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Neighbors

Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- Euclidean. (default) The distance between two points is computed by joining them with a straight line.
- Manhattan. The distance between two points is calculated as the sum of the absolute differences between their co-ordinates.
- Canberra. Similar to Manhattan distance, but more sensitive to data points closer to the origin.
- Maximum. The distance between two points is calculated as the greatest of their differences along any coordinate dimension.

Number of Nearest Neighbors (k). The number of nearest neighbors for a particular case. Note that using a greater number of neighbors will not necessarily result in a more accurate model.

The choice of k controls the balance between the prevention of overfitting (this may be important, particularly for "noisy" data) and resolution (yielding different predictions for similar instances). You will usually have to adjust the value of k for each data set, with typical values ranging from 1 to several dozen.

Enhance Performance and Accuracy

Standardize measurements before calculating distance. If selected, this option standardizes the measurements for continuous input fields before calculating the distance values.

Use coresets to increase performance for large datasets. If selected, this option uses core set sampling to speed up the calculation when large data sets are involved.

IBM Data WH KNN Model Options - Scoring Options

On the Model Options - Scoring Options tab, you can set the default value for a scoring option, and assign relative weights to individual classes.

Make Available for Scoring

Include input fields. Specifies whether the input fields are included in scoring by default.

Class Weights

Use this option if you want to change the relative importance of individual classes in building the model.

Note: This option is enabled only if you are using KNN for classification. If you are performing regression (that is, if the target field type is Continuous), the option is disabled.

The default is to assign a value of 1 to all classes, making them equally weighted. By specifying different numerical weights for different class labels, you instruct the algorithm to weight the training sets of particular classes accordingly.

To change a weight, double-click it in the Weight column and make the changes you want.

Value. The set of class labels, derived from the possible values of the target field.

Weight. The weighting to be assigned to a particular class. Assigning a higher weight to a class makes the model more sensitive to that class relative to the other classes.

IBM Data WH K-Means

The K-Means node implements the k -means algorithm, which provides a method of cluster analysis. You can use this node to cluster a data set into distinct groups.

The algorithm is a distance-based clustering algorithm that relies on a distance metric (function) to measure the similarity between data points. The data points are assigned to the nearest cluster according to the distance metric used.

The algorithm operates by performing several iterations of the same basic process, in which each training instance is assigned to the closest cluster (with respect to the specified distance function, applied to the instance and cluster center). All cluster centers are then recalculated as

the mean attribute value vectors of the instances assigned to particular clusters.

- [IBM Data WH K-Means Field Options](#)
 - [IBM Data WH K-Means Build Options Tab](#)
-

IBM Data WH K-Means Field Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the All button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM Data WH K-Means Build Options Tab

By setting the build options, you can customize the build of the model for your own purposes.

If you want to build a model with the default options, click Run.

Distance measure. This parameter defines the method of measure for the distance between data points. Greater distances indicate greater dissimilarities. Select one of the following options:

- **Euclidean.** The Euclidean measure is the straight-line distance between two data points.
- **Normalized Euclidean.** The Normalized Euclidean measure is similar to the Euclidean measure but it is normalized by the squared standard deviation. Unlike the Euclidean measure, the Normalized Euclidean measure is also scale-invariant.
- **Mahalanobis.** The Mahalanobis measure is a generalized Euclidean measure that takes correlations of input data into account. Like the Normalized Euclidean measure, the Mahalanobis measure is scale-invariant.
- **Manhattan.** The Manhattan measure is the distance between two data points that is calculated as the sum of the absolute differences between their coordinates.
- **Canberra.** The Canberra measure is similar to the Manhattan measure but it is more sensitive to data points that are closer to the origin.
- **Maximum.** The Maximum measure is the distance between two data points that is calculated as the greatest of their differences along any coordinate dimension.

Number of clusters. This parameter defines the number of clusters to be created.

Maximum number of iterations. The algorithm does several iterations of the same process. This parameter defines the number of iterations after which model training stops.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- **All.** All column-related statistics and all value-related statistics are included.
Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify None.
- **Columns.** Column-related statistics are included.
- **None.** Only statistics that are required to score the model are included.

Replicate results. Select this check box if you want to set a random seed to replicate analyses. You can specify an integer, or you can create a pseudo-random integer by clicking Generate.

IBM Data WH Naive Bayes

Naïve Bayes is a well-known algorithm for classification problems. The model is termed *naïve* because it treats all proposed prediction variables as being independent of one another. Naïve Bayes is a fast, scalable algorithm that calculates conditional probabilities for combinations of attributes and the target attribute. From the training data, an independent probability is established. This probability gives the likelihood of each target class, given the occurrence of each value category from each input variable.

Netezza Bayes Net

A Bayesian network is a model that displays variables in a data set and the probabilistic, or conditional, independencies between them. Using the Netezza Bayes Net node, you can build a probability model by combining observed and recorded evidence with common-sense real-world knowledge to establish the likelihood of occurrences by using seemingly unlinked attributes.

- [Netezza Bayes Net Field Options](#)
 - [Netezza Bayes Net Build Options](#)
-

Netezza Bayes Net Field Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

For this node, the target field is needed only for scoring, so it is not displayed on this tab. You can set or change the target on a Type node, on the Model Options tab of this node, or on the Settings tab of the model nugget. See the topic [Netezza Bayes Net Nugget - Settings Tab](#) for more information.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the All button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

Related information

- [Netezza Bayes Net](#)
 - [Netezza Bayes Net Model Nuggets](#)
-

Netezza Bayes Net Build Options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

Base index. The numeric identifier to be assigned to the first attribute (input field) for easier internal management.

Sample size. The size of the sample to take if the number of attributes is so large that it would cause an unacceptably long processing time.

Display additional information during execution. If this box is checked (default), additional progress information is displayed in a message dialog box.

Related information

- [Netezza Bayes Net](#)
 - [Netezza Bayes Net Model Nuggets](#)
-

Netezza Time Series

A **time series** is a sequence of numerical data values, measured at successive (though not necessarily regular) points in time--for example, daily stock prices or weekly sales data. Analyzing such data can be useful, for example, in highlighting behavior such as trends and seasonality (a repeating pattern), and in predicting future behavior from past events.

Netezza Time Series supports the following time series algorithms.

- spectral analysis
- exponential smoothing
- AutoRegressive Integrated Moving Average (ARIMA)
- seasonal trend decomposition

These algorithms break a time series down into a trend and a seasonal component. These components are then analyzed in order to build a model that can be used for prediction.

Spectral analysis is used to identify periodic behavior in time series. For time series composed of multiple underlying periodicities or when a considerable amount of random noise is present in the data, spectral analysis provides the clearest means of identifying periodic components. This method detects the frequencies of periodic behavior by transforming the series from the time domain into a series of the frequency domain.

Exponential smoothing is a method of forecasting that uses weighted values of previous series observations to predict future values. With exponential smoothing, the influence of observations decreases over time in an exponential way. This method forecasts one point at a time, adjusting its forecasts as new data comes in, taking into account addition, trend, and seasonality.

ARIMA models provide more sophisticated methods for modeling trend and seasonal components than do exponential smoothing models. This method involves explicitly specifying autoregressive and moving average orders as well as the degree of differencing.

Note: In practical terms, ARIMA models are most useful if you want to include predictors that may help to explain the behavior of the series being forecast, such as the number of catalogs mailed or the number of hits to a company Web page. Exponential smoothing models describe the behavior of the time series without attempting to explain why it behaves as it does.

Seasonal trend decomposition removes periodic behavior from the time series in order to perform a trend analysis and then selects a basic shape for the trend, such as a quadratic function. These basic shapes have a number of parameters whose values are determined so as to minimize the mean squared error of the residuals (that is, the differences between the fitted and observed values of the time series).

- [Interpolation of Values in Netezza Time Series](#)
- [Netezza Time Series Field Options](#)
- [Netezza Time Series Build Options](#)
- [Netezza Time Series Model Options](#)

Interpolation of Values in Netezza Time Series

Interpolation is the process of estimating and inserting missing values in time series data.

If the intervals of the time series are regular but some values are simply not present, the missing values can be estimated using linear interpolation. Consider the following series of monthly passenger arrivals at an airport terminal.

Table 1. Monthly arrivals at a passenger terminal

Month	Passengers
3	3,500,000
4	3,900,000
5	-
6	3,400,000
7	4,500,000
8	3,900,000
9	5,800,000
10	6,000,000

In this case, linear interpolation would estimate the missing value for month 5 as 3,650,000 (the mid-point between months 4 and 6).

Irregular intervals are handled differently. Consider the following series of temperature readings.

Table 2. Temperature readings

Date	Time	Temperature
2011-07-24	7:00	57
2011-07-24	14:00	75
2011-07-24	21:00	72
2011-07-25	7:15	59

Date	Time	Temperature
2011-07-25	14:00	77
2011-07-25	20:55	74
2011-07-27	7:00	60
2011-07-27	14:00	78
2011-07-27	22:00	74

Here we have readings taken at three points during three days, but at various times, only some of which are common between days. In addition, only two of the days are consecutive.

This situation can be handled in one of two ways: calculating aggregates, or determining a step size.

Aggregates might be daily aggregates calculated according to a formula based on semantic knowledge of the data. Doing so could result in the following data set.

Table 3. Temperature readings (aggregated)

Date	Time	Temperature
2011-07-24	24:00	69
2011-07-25	24:00	71
2011-07-26	24:00	null
2011-07-27	24:00	72

Alternatively, the algorithm can treat the series as a distinct series and determine a suitable step size. In this case, the step size determined by the algorithm might be 8 hours, resulting in the following.

Table 4. Temperature readings with step size calculated

Date	Time	Temperature
2011-07-24	6:00	
2011-07-24	14:00	75
2011-07-24	22:00	
2011-07-25	6:00	
2011-07-25	14:00	77
2011-07-25	22:00	
2011-07-26	6:00	
2011-07-26	14:00	
2011-07-26	22:00	
2011-07-27	6:00	
2011-07-27	14:00	78
2011-07-27	22:00	74

Here, only four readings correspond to the original measurements, but with the help of the other known values from the original series, the missing values can again be calculated by interpolation.

Netezza Time Series Field Options

On the Fields tab, you specify roles for the input fields in the source data.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Target. Choose one field as the target for the prediction. This must be a field with a measurement level of Continuous.

(Predictor) Time Points. (required) The input field containing the date or time values for the time series. This must be a field with a measurement level of Continuous or Categorical and a data storage type of Date, Time, Timestamp, or Numeric. The data storage type of the field you specify here also defines the input type for some fields on other tabs of this modeling node.

(Predictor) Time Series IDs (By). A field containing time series IDs; use this if the input contains more than one time series.

Related information

- [Netezza Time Series](#)
- [Netezza Time Series Model Nugget](#)

Netezza Time Series Build Options

There are two levels of build options:

- Basic - settings for the algorithm choice, interpolation, and time range to be used.
- Advanced - settings for forecasting

This section describes the basic options.

The Build Options tab is where you set all the options for building the model. You can, of course, just click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

Algorithm

These are the settings relating to the time series algorithm to be used.

Algorithm Name. Choose the time series algorithm you want to use. The available algorithms are Spectral Analysis, Exponential Smoothing (default), ARIMA, or Seasonal Trend Decomposition. See the topic [Netezza Time Series](#) for more information.

Trend. (Exponential Smoothing only) Simple exponential smoothing does not perform well if the time series exhibits a trend. Use this field to specify the trend, if any, so that the algorithm can take account of it.

- **System Determined.** (default) The system attempts to find the optimal value for this parameter.
- **None(N).** The time series does not exhibit a trend.
- **Additive(A).** A trend that steadily increases over time.
- **Damped Additive(DA).** An additive trend that eventually disappears.
- **Multiplicative(M).** A trend that increases over time, typically more rapidly than a steady additive trend.
- **Damped Multiplicative(DM).** A multiplicative trend that eventually disappears.

Seasonality. (Exponential Smoothing only) Use this field to specify whether the time series exhibits any seasonal patterns in the data.

- **System Determined.** (default) The system attempts to find the optimal value for this parameter.
- **None(N).** The time series does not exhibit seasonal patterns.
- **Additive(A).** The pattern of seasonal fluctuations exhibits a steady upward trend over time.
- **Multiplicative(M).** Same as additive seasonality, but in addition the amplitude (the distance between the high and low points) of the seasonal fluctuations increases relative to the overall upward trend of the fluctuations.

Use system determined settings for ARIMA. (ARIMA only) Choose this option if you want the system to determine the settings for the ARIMA algorithm.

Specify. (ARIMA only) Choose this option and click the button to specify the ARIMA settings manually.

Interpolation

If the time series source data has missing values, choose a method for inserting estimated values to fill the gaps in the data. See the topic [Interpolation of Values in Netezza Time Series](#) for more information.

- **Linear.** Choose this method if the intervals of the time series are regular but some values are simply not present.
- **Exponential Splines.** Fits a smooth curve where the known data point values increase or decrease at a high rate.
- **Cubic Splines.** Fits a smooth curve to the known data points to estimate the missing values.

Time Range

Here you can choose whether to use the full range of data in the time series, or a contiguous subset of that data, to create the model. Valid input for these fields is defined by the data storage type of the field specified for Time Points on the Fields tab. See the topic [Netezza Time Series Field Options](#) for more information.

- **Use earliest and latest times available in data.** Choose this option if you want to use the full range of the time series data.
- **Specify time window.** Choose this option if you want to use only a portion of the time series. Use the Earliest time (from) and Latest time (to) fields to specify the boundaries.
- [**ARIMA Structure**](#)
- [**Netezza Time Series Build Options - Advanced**](#)

Related information

- [Netezza Time Series](#)
- [Netezza Time Series Model Nugget](#)

ARIMA Structure

Specify the values of the various non-seasonal and seasonal components of the ARIMA model. In each case, set the operator to = (equal to) or <= (less than or equal to), then specify the value in the adjacent field. Values must be non-negative integers specifying the degrees.

Non seasonal. The values for the various nonseasonal components of the model.

- **Degrees of autocorrelation (p).** The number of autoregressive orders in the model. Autoregressive orders specify which previous values from the series are used to predict current values. For example, an autoregressive order of 2 specifies that the value of the series two time periods in the past be used to predict the current value.
- **Derivation (d).** Specifies the order of differencing applied to the series before estimating models. Differencing is necessary when trends are present (series with trends are typically nonstationary and ARIMA modeling assumes stationarity) and is used to remove their effect. The order of differencing corresponds to the degree of series trend--first-order differencing accounts for linear trends, second-order differencing accounts for quadratic trends, and so on.
- **Moving average (q).** The number of moving average orders in the model. Moving average orders specify how deviations from the series mean for previous values are used to predict current values. For example, moving-average orders of 1 and 2 specify that deviations from the mean value of the series from each of the last two time periods be considered when predicting current values of the series.

Seasonal. Seasonal autocorrelation (SP), derivation (SD), and moving average (SQ) components play the same roles as their nonseasonal counterparts. For seasonal orders, however, current series values are affected by previous series values separated by one or more seasonal periods. For example, for monthly data (seasonal period of 12), a seasonal order of 1 means that the current series value is affected by the series value 12 periods prior to the current one. A seasonal order of 1, for monthly data, is then the same as specifying a nonseasonal order of 12.

The seasonal settings are considered only if seasonality is detected in the data, or if you specify Period settings on the Advanced tab.

Netezza Time Series Build Options - Advanced

You can use the advanced settings to specify options for forecasting.

Use system determined settings for model build options. Choose this option if you want the system to determine the advanced settings.

Specify. Choose this option if you want to specify the advanced options manually. (The option is not available if the algorithm is Spectral Analysis.)

- **Period/Units for period.** The period of time after which some characteristic behavior of the time series repeats itself. For example, for a time series of weekly sales figures you would specify 1 for the period and Weeks for the units. Period must be a non-negative integer; Units for period can be one of Milliseconds, Seconds, Minutes, Hours, Days, Weeks, Quarters, or Years. Do not set Units for period if Period is not set, or if the time type is not numeric. However, if you specify Period, you must also specify Units for period.

Settings for forecasting. You can choose to make forecasts up to a particular point in time, or at specific time points. Valid input for these fields is defined by the data storage type of the field specified for Time Points on the Fields tab. See the topic [Netezza Time Series Field Options](#) for more information.

- **Forecast horizon.** Choose this option if you want to specify only an end point for forecasting. Forecasts will be made up to this point in time.
- **Forecast times.** Choose this option to specify one or more points in time at which to make forecasts. Click Add to add a new row to the table of time points. To delete a row, select the row and click Delete.

Netezza Time Series Model Options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also set default values for the model output options.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Make Available for Scoring. You can set the default values here for the scoring options that appear on the dialog for the model nugget.

- **Include historical values in outcome.** By default, the model output does not include the historical data values (the ones used to make the prediction). Select this check box to include these values.
- **Include interpolated values in outcome.** If you choose to include historical values in the output, select this box if you also want to include the interpolated values, if any. Note that interpolation works only on historical data, so this box is unavailable if Include historical values in outcome is not selected. See the topic [Interpolation of Values in Netezza Time Series](#) for more information.

IBM Data WH TwoStep

The TwoStep node implements the TwoStep algorithm that provides a method to cluster data over large data sets.

You can use this node to cluster data while available resources, for example, memory and time constraints, are considered.

The TwoStep algorithm is a database-mining algorithm that clusters data in the following way:

1. A clustering feature (CF) tree is created. This high-balanced tree stores clustering features for hierarchical clustering where similar input records become part of the same tree nodes.
 2. The leaves of the CF tree are clustered hierarchically in-memory to generate the final clustering result. The best number of clusters is determined automatically. If you specify a maximum number of clusters, the best number of clusters within the specified limit is determined.
 3. The clustering result is refined in a second step where an algorithm that is similar to the K-Means algorithm is applied to the data.
- [IBM Data WH TwoStep Field Options](#)
 - [IBM Data WH TwoStep Build Options](#)

IBM Data WH TwoStep Field Options

By setting the field options, you can specify to use the field role settings that are defined in upstream nodes. You can also make the field assignments manually.

Select an item. Choose this option to use the role settings from an upstream Type node or from the Types tab of an upstream source node. Role settings are, for example, targets and predictors.

Use custom field assignments. Choose this option if you want to assign targets, predictors, and other roles manually.

Fields. Use the arrows to assign items manually from this list to the role fields on the right. The icons indicate the valid measurement levels for each role field.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM Data WH TwoStep Build Options

By setting the build options, you can customize the build of the model for your own purposes.

If you want to build a model with the default options, click Run.

Distance measure. This parameter defines the method of measure for the distance between data points. Greater distances indicate greater dissimilarities. The options are:

- **Log-likelihood.** The likelihood measure places a probability distribution on the variables. Continuous variables are assumed to be normally distributed, while categorical variables are assumed to be multinomial. All variables are assumed to be independent.
- **Euclidean.** The Euclidean measure is the straight-line distance between two data points.
- **Normalized Euclidean.** The Normalized Euclidean measure is similar to the Euclidean measure but it is normalized by the squared standard deviation. Unlike the Euclidean measure, the Normalized Euclidean measure is also scale-invariant.

Cluster Number. This parameter defines the number of clusters to be created. The options are:

- **Automatically calculate number of clusters.** The number of clusters is calculated automatically. You can specify the maximum number of clusters in the Maximum field.
- **Specify number of clusters.** Specify how many clusters should be created.

Statistics. This parameter defines how many statistics are included in the model. The options are:

- **All.** All column-related statistics and all value-related statistics are included.
Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify None.
- **Columns.** Column-related statistics are included.
- **None.** Only statistics that are required to score the model are included.

Replicate results. Select this check box if you want to set a random seed to replicate analyses. You can specify an integer, or you can create a pseudo-random integer by clicking Generate.

IBM Data WH PCA

Principal component analysis (PCA) is a powerful data-reduction technique designed to reduce the complexity of data. PCA finds linear combinations of the input fields that do the best job of capturing the variance in the entire set of fields, where the components are orthogonal to (not correlated with) each other. The goal is to find a small number of derived fields (the principal components) that effectively summarize the information in the original set of input fields.

Note: An error may occur when scoring the model if lowercase field names are used. This is a known Db2 Data Warehouse defect, with the workaround being to rename all the fields to uppercase before scoring.

- [IBM Data WH PCA Field Options](#)
 - [IBM Data WH PCA Build Options](#)
-

IBM Data WH PCA Field Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the All button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM Data WH PCA Build Options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the Run button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

Center data before computing PCA. If checked (default), this option performs data centering (also known as "mean subtraction") before the analysis. Data centering is necessary to ensure that the first principal component describes the direction of maximum variance, otherwise the component might correspond more closely to the mean of the data. You would normally uncheck this option only for performance improvement if the data had already been prepared in this way.

Perform data scaling before computing PCA. This option performs data scaling before the analysis. Doing so can make the analysis less arbitrary when different variables are measured in different units. In its simplest form data scaling can be achieved by dividing each variable by its standard variation.

Use less accurate but faster method to compute PCA. This option causes the algorithm to use a less accurate but faster method (forceEigensolve) of finding the principal components.

Managing IBM Data WH and Netezza Models

IBM Data Warehouse and IBM® Netezza® Analytics models are added to the canvas and the Models palette in the same way as other IBM SPSS® Modeler models, and can be used in much the same way. However, there are a few important differences, given that each IBM Data Warehouse or IBM Netezza Analytics model created in IBM SPSS Modeler actually references a model stored on a database server. Thus for a stream to function correctly, it must connect to the database where the model was created, and the model table must not have been changed by an external process.

- [Scoring IBM Data Warehouse and IBM Netezza Analytics models](#)
 - [IBM Data WH and Netezza model nugget Server tab](#)
 - [IBM Data WH Decision Tree Model Nuggets](#)
 - [IBM Data WH K-Means Model Nugget](#)
 - [Netezza Bayes Net Model Nuggets](#)
 - [IBM Data WH Naive Bayes Model Nuggets](#)
 - [IBM Data WH KNN Model Nuggets](#)
 - [Netezza Divisive Clustering Model Nuggets](#)
 - [IBM Data WH PCA Model Nuggets](#)
 - [Netezza Regression Tree Model Nuggets](#)
 - [IBM Data WH Linear Regression Model Nuggets](#)
 - [Netezza Time Series Model Nugget](#)
 - [IBM Data WH Generalized Linear Model Nugget](#)
 - [IBM Data WH TwoStep Model Nugget](#)
-

Scoring IBM Data Warehouse and IBM® Netezza® Analytics models

Models are represented on the canvas by a gold model nugget icon. The main purpose of a nugget is for scoring data to generate predictions, or to allow further analysis of the model properties. Scores are added in the form of one or more extra data fields that can be made visible by attaching a Table node to the nugget and running that branch of the stream, as described later in this section. Some nugget dialog boxes, such as those for Decision Tree or Regression Tree, additionally have a Model tab that provides a visual representation of the model.

The extra fields are distinguished by the prefix \$<id>- added to the name of the target field, where <id> depends on the model, and identifies the type of information being added. The different identifiers are described in the topics for each model nugget.

To view the scores, complete the following steps:

1. Attach a Table node to the model nugget.
 2. Open the Table node.
 3. Click Run.
 4. Scroll to the right of the table output window to view the extra fields and their scores.
-

IBM Data WH and Netezza model nugget Server tab

On the Server tab, you can set server options for scoring the model. You can either continue to use a server connection that was specified upstream, or you can move the data to another database that you specify here.

IBM Data Warehouse Server Details. Here you specify the connection details for the database you want to use for the model.

- Use upstream connection. (default) Uses the connection details specified in an upstream node, for example the Database source node. This option works only if all upstream nodes are able to use SQL pushback. In this case there is no need to move the data out of the database, as the SQL fully implements all of the upstream nodes.
- Move data to connection. Moves the data to the database you specify here. Doing so allows modeling to work if the data is in another IBM Data Warehouse database, or a database from another vendor, or even if the data is in a flat file. In addition, data is moved back to the database specified here if the data has been extracted because a node did not perform SQL pushback. Click the Edit button to browse for and select a connection.
CAUTION:
IBM® Netezza® Analytics and IBM Data Warehouse is typically used with very large data sets. Transferring large amounts of data between databases, or out of and back into a database, can be very time-consuming and should be avoided where possible.

Model name. The name of the model. The name is shown for your information only; you cannot change it here.

IBM Data WH Decision Tree Model Nuggets

The Decision Tree model nugget displays the output from the modeling operation, and also enables you to set some options for scoring the model.

When you run a stream containing a Decision Tree model nugget, by default the node adds one new field, the name of which is derived from the target name.

Table 1. Model-scoring field for Decision Tree

Name of Added Field	Meaning
---------------------	---------

Name of Added Field	Meaning
<code>\$I-target_name</code>	Predicted value for current record.

If you select the option Compute probabilities of assigned classes for scoring records on either the modeling node or the model nugget and run the stream, a further field is added.

Table 2. Model-scoring field for Decision Tree - additional

Name of Added Field	Meaning
<code>\$IP-target_name</code>	Confidence value (from 0.0 to 1.0) for the prediction.

- [IBM Data WH Decision Tree Nugget - Model Tab](#)
- [IBM Data WH Decision Tree Nugget - Settings Tab](#)
- [IBM Data WH Decision Tree Nugget - Viewer Tab](#)

IBM Data WH Decision Tree Nugget - Model Tab

The Model tab shows the Predictor Importance of the decision tree model in graphical format. The length of the bar represents the importance of the predictor.

Note: When you are working with IBM® Netezza® Analytics Version 2.x or previous, the content of the decision tree model is shown in textual format only.

For these versions, the following information is shown:

- Each line of text corresponds to a node or a leaf.
- The indentation reflects the tree level.
- For a node, the split condition is displayed.
- For a leaf, the assigned class label is shown.

IBM Data WH Decision Tree Nugget - Settings Tab

The Settings tab enables you to set some options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Compute probabilities of assigned classes for scoring records. (Decision Tree and Naive Bayes only) If selected, this option means that the extra modeling fields include a confidence (that is, a probability) field as well as the prediction field. If you clear this check box, only the prediction field is produced.

Use deterministic input data. If selected, this option ensures that any Netezza algorithm that runs multiple passes of the same view will use the same set of data for each pass. If you clear this check box to show that non-deterministic data is being used a temporary table is created to hold the data output for processing, such as that produced by a partition node; this table is deleted after the model is created.

IBM Data WH Decision Tree Nugget - Viewer Tab

The Viewer tab shows a tree presentation of the tree model in the same way as the SPSS® Modeler does for its decision tree model.

Note: If the model is built with IBM® Netezza® Analytics Version 2.x or previous, the Viewer tab is empty.

IBM Data WH K-Means Model Nugget

K-Means model nuggets contain all of the information captured by the clustering model, as well as information about the training data and the estimation process.

When you run a stream containing a K-Means model nugget, the node adds two new fields containing the cluster membership and distance from the assigned cluster center for that record. The new field with the name `$KM-K-Means` is for the cluster membership and the new field with the name `$KMD-K-Means` is for the distance from the cluster center.

- [IBM Data WH K-Means Nugget - Model Tab](#)

- [IBM Data WH K-Means Nugget - Settings Tab](#)
-

IBM Data WH K-Means Nugget - Model Tab

The Model tab contains various graphic views that show summary statistics and distributions for fields of clusters. You can export the data from the model, or you can export the view as a graphic.

When you are working with IBM® Netezza® Analytics Version 2.x or previous, or when you build the model with Mahalanobis as the distance measure, the content of the K-Means models is shown in textual format only.

For these versions, the following information is shown:

- **Summary Statistics.** For both the smallest and the largest cluster, summary statistics shows the number of records. Summary statistics also shows the percentage of the data set that is taken up by these clusters. The list also shows the size ratio of the largest cluster to the smallest.
 - **Clustering Summary.** The clustering summary lists the clusters that are created by the algorithm. For each cluster, the table shows the number of records in that cluster, together with the mean distance from the cluster center for those records.
-

IBM Data WH K-Means Nugget - Settings Tab

The Settings tab enables you to set some options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- Euclidean. (default) The distance between two points is computed by joining them with a straight line.
 - Manhattan. The distance between two points is calculated as the sum of the absolute differences between their co-ordinates.
 - Canberra. Similar to Manhattan distance, but more sensitive to data points closer to the origin.
 - Maximum. The distance between two points is calculated as the greatest of their differences along any coordinate dimension.
-

Netezza Bayes Net Model Nuggets

The Bayes Net model nugget provides a means of setting options for scoring the model.

When you run a stream containing a Bayes Net model nugget, the node adds one new field, the name of which is derived from the target name.

Table 1. Model-scoring field for Bayes Net

Name of Added Field	Meaning
\$BN-target_name	Predicted value for current record.

You can view the extra field by attaching a Table node to the model nugget and running the Table node.

- [Netezza Bayes Net Nugget - Settings Tab](#)
-

Related information

- [Scoring IBM Data Warehouse and IBM Netezza Analytics models](#)
 - [Netezza Bayes Net](#)
-

Netezza Bayes Net Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Target. If you want to score a target field that is different from the current target, choose the new target here.

Record ID. If no Record ID field is specified, choose the field to use here.

Type of prediction. The variation of the prediction algorithm that you want to use:

- **Best (most correlated neighbor).** (default) Uses the most correlated neighbor node.
- **Neighbors (weighted prediction of neighbors).** Uses a weighted prediction of all neighbor nodes.
- **NN-neighbors (non-null neighbors).** Same as the previous option, except that it ignores nodes with null values (that is, nodes corresponding to attributes that have missing values for the instance for which the prediction is calculated).

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Related information

- [Scoring IBM Data Warehouse and IBM Netezza Analytics models](#)
 - [Netezza Bayes Net](#)
-

IBM Data WH Naive Bayes Model Nuggets

The Naive Bayes model nugget provides a means of setting options for scoring the model.

When you run a stream containing a Naive Bayes model nugget, by default the node adds one new field, the name of which is derived from the target name.

Table 1. Model-scoring field for Naive Bayes - default

Name of Added Field	Meaning
\$I-target_name	Predicted value for current record.

If you select the option Compute probabilities of assigned classes for scoring records on either the modeling node or the model nugget and run the stream, two further fields are added.

Table 2. Model-scoring fields for Naive Bayes - additional

Name of Added Field	Meaning
\$IP-target_name	The Bayesian numerator of the class for the instance (that is, the product of the prior class probability and the conditional instance attribute value probabilities).
\$ILP-target_name	The natural logarithm of the latter.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

- [IBM Data WH Naive Bayes Nugget - Settings Tab](#)
-

IBM Data WH Naive Bayes Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Compute probabilities of assigned classes for scoring records. (Decision Tree and Naive Bayes only) If selected, this option means that the extra modeling fields include a confidence (that is, a probability) field as well as the prediction field. If you clear this check box, only the prediction field is produced.

Improve probability accuracy for small or heavily unbalanced datasets. When computing probabilities, this option invokes the *m*-estimation technique for avoiding zero probabilities during estimation. This kind of estimation of probabilities may be slower but can give better results for small or heavily unbalanced datasets.

IBM Data WH KNN Model Nuggets

The KNN model nugget provides a means of setting options for scoring the model.

When you run a stream containing a KNN model nugget, the node adds one new field, the name of which is derived from the target name.

Table 1. Model-scoring field for KNN

Name of Added Field	Meaning
\$KNN-target_name	Predicted value for current record.

You can view the extra field by attaching a Table node to the model nugget and running the Table node.

- [IBM Data WH KNN Nugget - Settings Tab](#)

IBM Data WH KNN Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- Euclidean. (default) The distance between two points is computed by joining them with a straight line.
- Manhattan. The distance between two points is calculated as the sum of the absolute differences between their co-ordinates.
- Canberra. Similar to Manhattan distance, but more sensitive to data points closer to the origin.
- Maximum. The distance between two points is calculated as the greatest of their differences along any coordinate dimension.

Number of Nearest Neighbors (k). The number of nearest neighbors for a particular case. Note that using a greater number of neighbors will not necessarily result in a more accurate model.

The choice of k controls the balance between the prevention of overfitting (this may be important, particularly for "noisy" data) and resolution (yielding different predictions for similar instances). You will usually have to adjust the value of k for each data set, with typical values ranging from 1 to several dozen.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Standardize measurements before calculating distance. If selected, this option standardizes the measurements for continuous input fields before calculating the distance values.

Use coresets to increase performance for large datasets. If selected, this option uses core set sampling to speed up the calculation when large data sets are involved.

Netezza Divisive Clustering Model Nuggets

The Divisive Clustering model nugget provides a means of setting options for scoring the model.

When you run a stream containing a Divisive Clustering model nugget, the node adds two new fields, the names of which are derived from the target name.

Table 1. Model-scoring fields for Divisive Clustering

Name of Added Field	Meaning
\$DC-target_name	Identifier of subcluster to which current record is assigned.
\$DCD-target_name	Distance from subcluster center for current record.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

- [Netezza Divisive Clustering Nugget - Settings Tab](#)

Related information

- [Scoring IBM Data Warehouse and IBM Netezza Analytics models](#)
- [Netezza Divisive Clustering](#)

Netezza Divisive Clustering Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- Euclidean. (default) The distance between two points is computed by joining them with a straight line.
- Manhattan. The distance between two points is calculated as the sum of the absolute differences between their co-ordinates.
- Canberra. Similar to Manhattan distance, but more sensitive to data points closer to the origin.
- Maximum. The distance between two points is calculated as the greatest of their differences along any coordinate dimension.

Applied hierarchy level. The level of hierarchy that should be applied to the data.

Related information

- [Scoring IBM Data Warehouse and IBM Netezza Analytics models](#)
 - [Netezza Divisive Clustering](#)
-

IBM Data WH PCA Model Nuggets

The PCA model nugget provides a means of setting options for scoring the model.

When you run a stream containing a PCA model nugget, by default the node adds one new field, the name of which is derived from the target name.

Table 1. Model-scoring field for PCA

Name of Added Field	Meaning
\$F-target_name	Predicted value for current record.

If you specify a value greater than 1 in the Number of principal components ... field on either the modeling node or the model nugget and run the stream, the node adds a new field for each component. In this case the field names are suffixed by *-n*, where *n* is the number of the component. For example, if your model is named *pca* and contains three components, the new fields would be named \$F-pca-1, \$F-pca-2, and \$F-pca-3.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

Note: An error may occur when scoring the model if lowercase field names are used. This is a known Db2 Data Warehouse defect, with the workaround being to rename all the fields to uppercase before scoring.

- [IBM Data WH PCA Nugget - Settings Tab](#)
-

IBM Data WH PCA Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Number of principal components to be used in projection. The number of principal components to which you want to reduce the data set. This value must not exceed the number of attributes (input fields).

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Netezza Regression Tree Model Nuggets

The Regression Tree model nugget provides a means of setting options for scoring the model.

When you run a stream containing a Regression Tree model nugget, by default the node adds one new field, the name of which is derived from the target name.

Table 1. Model-scoring field for Regression Tree

Name of Added Field	Meaning
\$I-target_name	Predicted value for current record.

If you select the option Compute estimated variance on either the modeling node or the model nugget and run the stream, a further field is added.

Table 2. Model-scoring field for Regression Tree - additional

Name of Added Field	Meaning
\$IV-target_name	Estimated variances of the predicted value.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

- [Netezza Regression Tree Nugget - Model Tab](#)
- [Netezza Regression Tree Nugget - Settings Tab](#)
- [Netezza Regression Tree Nugget - Viewer Tab](#)

Related information

- [Scoring IBM Data Warehouse and IBM Netezza Analytics models](#)
- [IBM Data WH Regression Tree](#)

Netezza Regression Tree Nugget - Model Tab

The Model tab shows the Predictor Importance of the regression tree model in graphical format. The length of the bar represents the importance of the predictor.

Note: When you are working with IBM® Netezza® Analytics Version 2.x or previous, the content of the regression tree model is shown in textual format only.

For these versions, the following information is shown:

- Each line of text corresponds to a node or a leaf.
- The indentation reflects the tree level.
- For a node, the split condition is displayed.
- For a leaf, the assigned class label is shown.

Related information

- [Scoring IBM Data Warehouse and IBM Netezza Analytics models](#)
- [IBM Data WH Regression Tree](#)

Netezza Regression Tree Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Compute estimated variance. Indicates whether the variances of assigned classes should be included in the output.

Related information

- [Scoring IBM Data Warehouse and IBM Netezza Analytics models](#)
- [IBM Data WH Regression Tree](#)

Netezza Regression Tree Nugget - Viewer Tab

The Viewer tab shows a tree presentation of the tree model in the same way as the SPSS® Modeler does for its regression tree model.

Note: If the model is built with IBM® Netezza® Analytics Version 2.x or previous, the Viewer tab is empty.

Related information

- [Scoring IBM Data Warehouse and IBM Netezza Analytics models](#)
 - [IBM Data WH Regression Tree](#)
-

IBM Data WH Linear Regression Model Nuggets

The Linear Regression model nugget provides a means of setting options for scoring the model.

When you run a stream containing a Linear Regression model nugget, the node adds one new field, the name of which is derived from the target name.

Table 1. Model-scoring field for Linear Regression

Name of Added Field	Meaning
\$LR-target_name	Predicted value for current record.

- [IBM Data WH Linear Regression Nugget - Settings Tab](#)
-

IBM Data WH Linear Regression Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Netezza Time Series Model Nugget

The model nugget provides access to the output of the time series modeling operation. The output consists of the following fields.

Table 1. Time Series model output fields

Field	Description
TSID	The identifier of the time series; the contents of the field specified for Time Series IDs on the Fields tab of the modeling node. See the topic Netezza Time Series Field Options for more information.
TIME	The time period within the current time series.
HISTORY	The historical data values (the ones used to make the prediction). This field is included only if the option Include historical values in outcome is selected on the Settings tab of the model nugget.
\$TS-INTERPOLATED	The interpolated values, where used. This field is included only if the option Include interpolated values in outcome is selected on the Settings tab of the model nugget. Interpolation is an option on the Build Options tab of the modeling node.
\$TS-FORECAST	The forecast values for the time series.

To view the model output, attach a Table node (from the Output tab of the node palette) to the model nugget and run the Table node.

- [Netezza Time Series Nugget - Settings tab](#)
-

Netezza Time Series Nugget - Settings tab

On the Settings tab you can specify options for customizing the model output.

Model Name. The name of the model, as specified on the Model Options tab of the modeling node.

The other options are the same as those on the Modeling Options tab of the modeling node.

IBM Data WH Generalized Linear Model Nugget

The model nugget provides access to the output of the modeling operation.

When you run a stream containing a Generalized Linear model nugget, the node adds a new field, the name of which is derived from the target name.

Table 1. Model-scoring field for Generalized Linear

Name of Added Field	Meaning
\$GLM-target_name	Predicted value for current record.

The Model tab displays various statistics relating to the model.

The output consists of the following fields.

Table 2. Output fields from Generalized Linear model

Output Field	Description
Parameter	The parameters (that is, the predictor variables) used by the model. These are the numerical and nominal columns, as well as the intercept (the constant term in the regression model).
Beta	The correlation coefficient (that is, the linear component of the model).
Std Error	The standard deviation for the beta.
Test	The test statistics used to evaluate the validity of the parameter.
p-value	The probability of an error when assuming that the parameter is significant.
Residuals Summary	
Residual Type	The type of residual of the prediction for which summary values are shown.
RSS	The value of the residual.
df	The degrees of freedom for the residual.
p-value	The probability of an error. A high value indicates a poorly-fitting model; a low value indicates a good fit.

- [IBM Data WH Generalized Linear Model Nugget - Settings tab](#)

IBM Data WH Generalized Linear Model Nugget - Settings tab

On the Settings tab you can customize the model output.

The option is the same as that shown for Scoring Options on the modeling node. See the topic [IBM Data WH Generalized Linear Model Options - Scoring Options](#) for more information.

IBM Data WH TwoStep Model Nugget

When you run a stream that contains a TwoStep model nugget, the node adds two new fields that contain the cluster membership and distance from the assigned cluster center for that record. The new field with the name \$TS-Twostep is for the cluster membership, and the new field with the name \$TSP-Twostep is for the distance from the cluster center.

- [IBM Data WH TwoStep Nugget - Model Tab](#)

IBM Data WH TwoStep Nugget - Model Tab

The Model tab contains various graphic views that show summary statistics and distributions for fields of clusters. You can export the data from the model, or you can export the view as a graphic.

Database modeling with IBM® Db2® for z/OS®

- [IBM SPSS Modeler and IBM Db2 for z/OS](#)
- [Requirements for integration with IBM Db2 for z/OS](#)
- [Enabling integration with IBM Db2 Analytics Accelerator for z/OS](#)
- [Building models with IBM Db2 for z/OS](#)
- [IBM Db2 for z/OS Models - K-Means](#)

- [IBM Db2 for z/OS models - Naive Bayes](#)
 - [IBM Db2 for z/OS Models - Decision Trees](#)
 - [IBM Db2 for z/OS models - Regression Tree](#)
 - [IBM Db2 for z/OS Models - Regression Tree Build Options - Tree Growth](#)
 - [IBM Db2 for z/OS models - Regression Tree build options - Tree Pruning](#)
 - [IBM Db2 for z/OS models - TwoStep](#)
 - [Managing IBM Db2 for z/OS Models](#)
-

IBM SPSS Modeler and IBM Db2 for z/OS

SPSS Modeler supports integration with Db2 for z/OS, which provides the ability to run advanced analytics on Db2 for z/OS servers. You can access these features through the SPSS Modeler graphical user interface and workflow-oriented development environment. This way, you can run the data mining algorithms directly in the Db2 for z/OS environment leveraging IBM Db2 Analytics Accelerator.

SPSS Modeler supports integration of the following algorithms from Db2 for z/OS.

- Decision Trees
 - K-Means
 - Naive Bayes
 - Regression Tree
 - TwoStep
-

Requirements for integration with IBM Db2 for z/OS

The following conditions are prerequisites for conducting in-database modeling by using Db2® for z/OS® and IBM® Db2 Analytics Accelerator for z/OS. To ensure that these conditions are met, you might need to consult with your database administrator. For detailed requirements, including supported versions, see the [Software Product Compatibility Reports](#).

- IBM SPSS® Modeler running in local mode or against an SPSS Modeler Server installation on Windows or UNIX
 - Db2 for z/OS together with Db2 Analytics Accelerator for z/OS
 - IBM SPSS Data Access Pack
 - On the server running SPSS Modeler Server, one of the following systems:
 - IBM Db2 Data Server Driver for ODBC and CLI
 - Any version of Db2 for Linux®, UNIX, and Windows with an ODBC data source that is configured for Db2 for z/OS
 - License for Db2 Connect for System z®
 - SQL generation and optimization enabled in SPSS Modeler
 - Db2 z/OS in-database mining requires either accelerator-only tables (AOT) or accelerated tables, and INZA support. IDAA INZA was introduced in IDAA 5.1. This means that the Db2 z/OS in-database mining nodes will not work with previous versions of IDAA. If you use an IDAA-enabled DSN in Modeler, the only tables that will be displayed in the list of tables returned in the Database source node using that DSN will be AOT or accelerated tables.
-

Enabling integration with IBM Db2 Analytics Accelerator for z/OS

Enabling integration with Db2® Analytics Accelerator for z/OS® consists of the following steps:

- Configuring Db2 for z/OS and Db2 Analytics Accelerator for z/OS
 - Creating an ODBC source
 - Enabling the integration of IBM® Db2 for z/OS in IBM SPSS® Modeler
 - Enabling SQL generation and optimization in SPSS Modeler
 - Enabling IBM SPSS Modeler Server Scoring Adapter for Db2 for z/OS
 - Configuring DSN using IBM Db2 Client in IBM SPSS Modeler
 - [Configuring IBM Db2 for z/OS and IBM Analytics Accelerator for z/OS](#)
 - [Creating an ODBC Source for IBM Db2 for z/OS and IBM Db2 Analytics Accelerator](#)
 - [Enabling the integration of IBM Db2 for z/OS in IBM SPSS Modeler](#)
 - [Enabling SQL Generation and Optimization](#)
 - [Configuring DSN using IBM Db2 Client in IBM SPSS Modeler](#)
-

Configuring IBM® Db2 for z/OS and IBM Analytics Accelerator for z/OS

How to configure Db2® for z/OS® and Analytics Accelerator for z/OS is described on the following website:

[Db2 Analytics Accelerator for z/OS](#).

Creating an ODBC Source for IBM Db2 for z/OS and IBM Db2 Analytics Accelerator

For information about how to enable a connection between Db2® for z/OS® and IBM® Db2 Analytics Accelerator, see the following websites:

- For version 4: [Db2 Analytics Accelerator for z/OS 4.1.0](#)
- For version 3: [Db2 Analytics Accelerator for z/OS 3.1.0](#)
- [Enabling query acceleration with IBM Db2 Analytics Accelerator for ODBC and JDBC applications without modifying the applications](#)
- [SQL error from ODBC driver when running a query in Db2 Analytics Accelerator for z/OS](#)

Enabling the integration of IBM® Db2 for z/OS® in IBM SPSS Modeler

To enable the integration of Db2® for z/OS in SPSS® Modeler, perform the following steps:

1. From the SPSS Modeler config directory, open the `odbc-db2-accelerator-names.cfg` file.
If the file does not exists, you must create it.

2. Add the names of all data sources and the names of all accelerators. For example:

```
dsn1, acceleratorname1  
dsn2, acceleratorname2
```

3. The default CCSID for accelerator only tables (AOT) is Unicode; to override this, modify the entries by adding encoding strings to the accelerator names. For example:

```
dsn1, acceleratorname1, EBCDIC  
dsn2, acceleratorname2, UNICODE
```

4. Save and close the `odbc-db2-accelerator-names.cfg` file, then open the `odbc-db2-custom-properties.cfg` file from the same directory.

5. SPSS Modeler uses SQL to set the IDAA registers. If required, you can override these entries by changing the SQL to the required values.
For example:

```
current_query_sql_acc, "SET CURRENT QUERY ACCELERATION = ELIGIBLE"  
current_get_archive_acc, "SET CURRENT GET_ACCEL_ARCHIVE = NO"
```

6. By default, SPSS Modeler uses SQL to create temporary tables for a database cache. If required, you can override this by specifying the expected database name. For example:

```
[OSZ]  
table_create_temp_sql_acc, 'CREATE TABLE <table-name> <(table-columns)> IN DATABASE  
NAME_OF_DATABASE_FOR_AOT'
```

7. By default, SPSS Modeler considers that SQL queries written in an ODBC source node are non-replayable, meaning that the query is considered to return different results when being executed multiple times. However, in some scenarios, this may prevent Modeler from generating SQL for downstream nodes and can be overridden by changing the relevant value to `Y`. For example:

```
assume_custom_sql_replayable, Y
```

8. From the SPSS Modeler main menu, click Tools > Options > Helper Applications.

9. Click the IBM Db2 for z/OS tab.

10. Select Enable IBM Db2 for z/OS Data Mining Integration and then click OK.

Note: You cannot view IDAA and non-IDAA tables at the same time in Modeler.

Enabling SQL Generation and Optimization

Because of the likelihood of working with very large data sets, for performance reasons you should enable the SQL generation and optimization options in IBM® SPSS® Modeler.

To configure SPSS Modeler, do the following steps:

1. From the IBM SPSS Modeler menus choose Tools > Stream Properties > Options
 2. Click the Optimization option in the navigation pane.
 3. Confirm that the Generate SQL option is enabled. This setting is required for database modeling to function.
 4. Select Optimize SQL Generation and Optimize other execution (not strictly required but strongly recommended for optimized performance).
-

Configuring DSN using IBM® Db2 Client in IBM SPSS Modeler

If required, to configure a data source name (DSN) using Db2® Client for Db2 in SPSS® Modeler, complete the following steps:

1. If not already installed, install Db2 Client on the operating system where Modeler Server is installed.
2. Using the **db2 catalog** command, catalog the database and add a new data source to the db2cli.ini file in Db2 Client. Be sure to point to the defined database alias.
3. Configure data access; detailed steps are available in the Modeler documentation.
For more information, see [Data Access](#).
4. Create a new ODBC data source in odbc.ini by referencing the database alias defined in step 2.
5. For Linux or UNIX users:
 - a. Ensure that the driver library libdb2o.so is used (instead of libdb2.so), and make sure 'DriverUnicodeType=1' is defined for the new data source.
 - b. In the IBM SPSS Data Access Pack installation, ensure that the library path of Db2 Client is added to odbc.sh.
 - c. Ensure that Modeler Server uses an ODBC Driver wrapper library with UTF-16 encoding (this is called 'libspssodbc_datadirect_utf16.so').
6. Make sure that the user who connects to Db2 has the necessary privileges to run the following query:

```
SELECT ACCELERATORNAME FROM SYSACCEL.SYSACCELERATORS
```

Building models with IBM® Db2 for z/OS

Each of the supported algorithms has a corresponding modeling node. You can access the Db2® for z/OS® modeling nodes from the Database Modeling tab on the nodes palette.

Data considerations

Fields in the data source can contain variables of various data types, depending on the modeling node. In SPSS® Modeler, data types are known as *measurement levels*. The Fields tab of the modeling node uses icons to indicate the permitted measurement level types for its input and target fields.

Target field. The target field is the field whose value you are trying to predict. Where a target can be specified, only one of the source data fields can be selected as the target field.

Record ID field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. If the source data does not include an ID field, you can create this field by means of a Derive node, as the following procedure shows.

1. Select the source node.
2. From the Field Ops tab on the nodes palette, double-click the Derive node.
3. Open the Derive node by double-clicking its icon on the canvas.
4. In the Derive field field, type (for example) *ID*.
5. In the Formula field, type @INDEX and click OK.
6. Connect the Derive node to the rest of the stream.

Handling null values

If the input data contains null values, use of some Db2 for z/OS nodes may result in error messages or long-running streams, so we recommend removing records containing null values. Use the following method.

1. Attach a Select node to the source node.
2. Set the Mode option of the Select node to Discard.
3. Enter the following in the Condition field:

```
@NULL(field1) [or @NULL(field2) [...] or @NULL(fieldN)])
```

Be sure to include every input field.

4. Connect the Select node to the rest of the stream.

Model output

It is possible for a stream containing a Db2 for z/OS modeling node to produce slightly different results each time it is run. This is because the order in which the node reads the source data is not always the same, as the data is read into temporary tables before model building. However, the differences produced by this effect are negligible.

General comments

- In SPSS Collaboration and Deployment Services, it is not possible to create scoring configurations using streams containing Db2 for z/OS modeling nodes.
 - PMML export or import is not possible for models created by the Db2 for z/OS nodes.
 - [IBM Db2 for z/OS models - Field options](#)
 - [IBM Db2 for z/OS Models - Server Options](#)
 - [IBM Db2 for z/OS models - Model options](#)
-

IBM® Db2® for z/OS® models - Field options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the All button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Target. Choose one field as the target for the prediction. For Generalized Linear models, see also the Trials field on this screen.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM Db2 for z/OS Models - Server Options

On the Server tab, you specify the Db2® for z/OS® system where the model is to be built.

- Use upstream connection. (default) Uses the connection details specified in an upstream node, for example the Database source node.
Note: This option works only if all upstream nodes are able to use SQL pushback. In this case, there is no need to move the data out of the database, as the SQL fully implements all of the upstream nodes.
- Move data to connection. Moves the data to the database you specify here. Doing so allows modeling to work if the data is in another IBM® database, or a database from another vendor, or even if the data is in a flat file. In addition, data is moved back to the database specified here if the data has been extracted because a node did not perform SQL pushback. Click the Edit button to browse for and select a connection.

Note: The ODBC data source name is effectively embedded in each SPSS® Modeler stream. If a stream that is created on one host is executed on a different host, the name of the data source must be the same on each host. Alternatively, a different data source can be selected on the Server tab in each source or modeling node.

IBM® Db2® for z/OS® models - Model options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Replace existing if the name has been used. If you select this check box, any existing model of the same name will be overwritten.

IBM® Db2® for z/OS® Models - K-Means

The K-Means node implements the *k*-means algorithm, which provides a method of cluster analysis. You can use this node to cluster a data set into distinct groups.

The algorithm is a distance-based clustering algorithm that relies on a distance metric (function) to measure the similarity between data points. The data points are assigned to the nearest cluster according to the distance metric used.

The algorithm operates by performing several iterations of the same basic process, in which each training instance is assigned to the closest cluster (with respect to the specified distance function, applied to the instance and cluster center). All cluster centers are then recalculated as the mean attribute value vectors of the instances assigned to particular clusters.

- [IBM Db2 for z/OS models - K-Means Field options](#)
- [IBM Db2 for z/OS Models - K-Means build options](#)

IBM® Db2® for z/OS® models - K-Means Field options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the All button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM® Db2® for z/OS® Models - K-Means build options

By setting the build options, you can customize the build of the model for your own purposes.

If you want to build a model with the default options, click Run.

Distance measure. This parameter defines the method of measure for the distance between data points. Greater distances indicate greater dissimilarities. Select one of the following options:

- Euclidean. The Euclidean measure is the straight-line distance between two data points.
- Normalized Euclidean. The Normalized Euclidean measure is similar to the Euclidean measure but it is normalized by the squared standard deviation. Unlike the Euclidean measure, the Normalized Euclidean measure is also scale-invariant.

Number of clusters. This parameter defines the number of clusters to be created.

Maximum number of iterations. The algorithm does several iterations of the same process. This parameter defines the number of iterations after which model training stops.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- All. All column-related statistics and all value-related statistics are included.
Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify None.
- Columns. Column-related statistics are included.
- None. Only statistics that are required to score the model are included.

Replicate results. Select this check box if you want to set a random seed to replicate analyses. You can specify an integer, or you can create a pseudo-random integer by clicking Generate.

IBM® Db2® for z/OS® models - Naive Bayes

Naive Bayes is a well-known algorithm for classification problems. The model is termed naïve because it treats all proposed prediction variables as being independent of one another. Naive Bayes is a fast, scalable algorithm that calculates conditional probabilities for combinations of attributes and the target attribute. From the training data, an independent probability is established. This probability gives the likelihood of each target class, given the occurrence of each value category from each input variable.

IBM® Db2® for z/OS® Models - Decision Trees

A decision tree is a hierarchical structure that represents a classification model. With a decision tree model, you can develop a classification system to predict or classify future observations from a set of training data. The classification takes the form of a tree structure in which the branches represent split points in the classification. The splits break the data down into subgroups recursively until a stopping point is reached. The tree nodes at the stopping points are known as *leaves*. Each leaf assigns a label, known as a *class label*, to the members of its subgroup, or class.

- [IBM Db2 for z/OS models - Decision Tree field options](#)
- [IBM Db2 for z/OS Models - Decision Tree Build Options](#)
- [IBM Db2 for z/OS Models - Decision Tree Node - Class Weights](#)
- [IBM Db2 for z/OS Models - Decision Tree Node - Tree Pruning](#)

IBM® Db2® for z/OS® models - Decision Tree field options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the All button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Target. Choose one field as the target for the prediction.

Record ID. The field that is to be used as the unique record identifier. The values of this field must be unique for each record (for example, customer ID numbers).

Instance Weight. Specifying a field here enables you to use instance weights (a weight per row of input data) instead of, or in addition to, the default, class weights (a weight per category for the target field). The field you specify here must be one that contains a numeric weight for each row of input data.

Predictors (Inputs). Select the input field or fields. This is similar to setting the field role to *Input* in a Type node.

IBM® Db2® for z/OS® Models - Decision Tree Build Options

The following build options are available for tree growth:

Growth Measure. These options control the way tree growth is measured.

- Impurity Measure. This measure evaluates the best place to split the tree. It is a measurement of the variability in a subgroup or segment of data. A low impurity measurement indicates a group where most members have similar values for the criterion or target field. The supported measurements are Entropy and Gini. These measurements are based on probabilities of category membership for the branch.
- Maximum tree depth. The maximum number of levels to which the tree can grow below the root node, that is, the number of times the sample is split recursively. The default value of this property is 10, and the maximal value that you can set for this property is 62. Note: If the viewer in the model nugget shows the textual representation of the model, a maximum of 12 levels of the tree is displayed.

Splitting Criteria. These options control when to stop splitting the tree.

- Minimum improvement for splits. The minimum amount by which impurity must be reduced before a new split is created in the tree. The goal of tree building is to create subgroups with similar output values to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the amount that is specified by the splitting criteria, the branch is not split.
- Minimum number of instances for a split. The minimum number of records that can be split. When fewer than this number of unsplittable records remain, no further splits are made. You can use this field to prevent the creation of small subgroups in the tree.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- All. All column-related statistics and all value-related statistics are included.
Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify None.
- Columns. Column-related statistics are included.
- None. Only statistics that are required to score the model are included.

IBM® Db2® for z/OS® Models - Decision Tree Node - Class Weights

Here you can assign weights to individual classes. The default is to assign a value of 1 to all classes, making them equally weighted. By specifying different numerical weights for different class labels, you instruct the algorithm to weight the training sets of particular classes accordingly.

To change a weight, double-click it in the Weight column and make the changes you want.

Value. The set of class labels, derived from the possible values of the target field.

Weight. The weighting to be assigned to a particular class. Assigning a higher weight to a class makes the model more sensitive to that class relative to the other classes.

You can use class weights in combination with instance weights.

IBM® Db2® for z/OS® Models - Decision Tree Node - Tree Pruning

You can use the pruning options to specify pruning criteria for the decision tree. The intention of pruning is to reduce the risk of overfitting by removing overgrown subgroups that do not improve the expected accuracy on new data.

Pruning measure. The default pruning measure, Accuracy, ensures that the estimated accuracy of the model remains within acceptable limits after removing a leaf from the tree. Use the alternative, Weighted Accuracy, if you want to take the class weights into account while applying pruning.

Data for pruning. You can use some or all of the training data to estimate the expected accuracy on new data. Alternatively, you can use a separate pruning dataset from a specified table for this purpose.

- **Use all training data.** This option (the default) uses all the training data to estimate the model accuracy.
- **Use % of training data for pruning.** Use this option to split the data into two sets, one for training and one for pruning, using the percentage specified here for the pruning data.
- Select Replicate results if you want to specify a random seed to ensure that the data is partitioned in the same way each time you run the stream. You can either specify an integer in the Seed used for pruning field, or click Generate, which will create a pseudo-random integer.
- **Use data from an existing table.** Specify the table name of a separate pruning data set for estimating model accuracy. Doing so is considered more reliable than using training data.

IBM® Db2® for z/OS® models - Regression Tree

A regression tree is a tree-based algorithm that splits a sample of cases repeatedly to derive subsets of the same kind, based on values of a numeric target field. As with decision trees, regression trees decompose the data into subsets in which the leaves of the tree correspond to sufficiently small or sufficiently uniform subsets. Splits are selected to decrease the dispersion of target attribute values, so that they can be reasonably well predicted by their mean values at leaves.

IBM® Db2® for z/OS® Models - Regression Tree Build Options - Tree Growth

You can set build options for tree growth and tree pruning.

The following build options are available for tree growth:

Maximum tree depth. The maximum number of levels to which the tree can grow below the root node, that is, the number of times the sample is split recursively. The default is 62, which is the maximum tree depth for modeling purposes.

Note: If the viewer in the model nugget shows the textual representation of the model, a maximum of 12 levels of the tree is displayed. Splitting Criteria. These options control when to stop splitting the tree.

- Split evaluation measure. This class evaluation measure evaluates the best place to split the tree.
Note: Currently, variance is the only possible option.
- Minimum improvement for splits. The minimum amount by which impurity must be reduced before a new split is created in the tree. The goal of tree building is to create subgroups with similar output values to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the amount that is specified by the splitting criteria, the branch is not split.
- Minimum number of instances for a split. The minimum number of records that can be split. When fewer than this number of unsplit records remain, no further splits are made. You can use this field to prevent the creation of small subgroups in the tree.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- All. All column-related statistics and all value-related statistics are included.
Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify None.
- Columns. Column-related statistics are included.
- None. Only statistics that are required to score the model are included.

IBM® Db2® for z/OS® models - Regression Tree build options - Tree Pruning

You can use the pruning options to specify pruning criteria for the regression tree. The intention of pruning is to reduce the risk of overfitting by removing overgrown subgroups that do not improve the expected accuracy on new data.

Pruning measure. The pruning measure ensures that the estimated accuracy of the model remains within acceptable limits after removing a leaf from the tree. You can select one of the following measures.

- mse. Mean squared error - (default) measures how close a fitted line is to the data points.
- r2. R-squared - measures the proportion of variation in the dependent variable explained by the regression model.
- Pearson. Pearson's correlation coefficient - measures the strength of relationship between linearly dependent variables that are normally distributed.
- Spearman. Spearman's correlation coefficient - detects nonlinear relationships that appear weak according to Pearson's correlation, but which may actually be strong.

Data for pruning. You can use some or all of the training data to estimate the expected accuracy on new data. Alternatively, you can use a separate pruning dataset from a specified table for this purpose.

- Use all training data. This option (the default) uses all the training data to estimate the model accuracy.
- Use % of training data for pruning. Use this option to split the data into two sets, one for training and one for pruning, using the percentage specified here for the pruning data.
Select Replicate results if you want to specify a random seed to ensure that the data is partitioned in the same way each time you run the stream. You can either specify an integer in the Seed used for pruning field, or click Generate, which will create a pseudo-random integer.
- Use data from an existing table. Specify the table name of a separate pruning dataset for estimating model accuracy. Doing so is considered more reliable than using training data.

IBM® Db2® for z/OS® models - TwoStep

The TwoStep node implements the TwoStep algorithm that provides a method to cluster data over large data sets.

You can use this node to cluster data while available resources, for example, memory and time constraints, are considered.

The TwoStep algorithm is a database-mining algorithm that clusters data in the following way:

1. A clustering feature (CF) tree is created. This high-balanced tree stores clustering features for hierarchical clustering where similar input records become part of the same tree nodes.
2. The leaves of the CF tree are clustered hierarchically in-memory to generate the final clustering result. The best number of clusters is determined automatically. If you specify a maximum number of clusters, the best number of clusters within the specified limit is

determined.

3. The clustering result is refined in a second step where an algorithm that is similar to the K-Means algorithm is applied to the data.

- [IBM Db2 for z/OS models - TwoStep field options](#)
 - [IBM Db2 for z/OS Models - TwoStep Build Options](#)
 - [IBM Db2 for z/OS Models - TwoStep nugget - Model tab](#)
-

IBM® Db2® for z/OS® models - TwoStep field options

By setting the field options, you can specify to use the field role settings that are defined in upstream nodes. You can also make the field assignments manually.

Select an item. Choose this option to use the role settings from an upstream Type node or from the Types tab of an upstream source node. Role settings are, for example, targets and predictors.

Use custom field assignments. Choose this option if you want to assign targets, predictors, and other roles manually.

Fields. Use the arrows to assign items manually from this list to the role fields on the right. The icons indicate the valid measurement levels for each role field.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM® Db2® for z/OS® Models - TwoStep Build Options

By setting the build options, you can customize the build of the model for your own purposes.

If you want to build a model with the default options, click Run.

Distance measure. This parameter defines the method of measure for the distance between data points. Greater distances indicate greater dissimilarities. The option is:

- Log-likelihood. The likelihood measure places a probability distribution on the variables. Continuous variables are assumed to be normally distributed, while categorical variables are assumed to be multinomial. All variables are assumed to be independent.

Cluster Number. This parameter defines the number of clusters to be created. The options are:

- Automatically calculate number of clusters. The number of clusters is calculated automatically. You can specify the maximum number of clusters in the Maximum field.
- Specify number of clusters. Specify how many clusters should be created.

Statistics. This parameter defines how many statistics are included in the model. The options are:

- All. All column-related statistics and all value-related statistics are included.
Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify None.
- Columns. Column-related statistics are included.
- None. Only statistics that are required to score the model are included.

Replicate results. Select this check box if you want to set a random seed to replicate analyses. You can specify an integer, or you can create a pseudo-random integer by clicking Generate.

IBM® Db2® for z/OS® Models - TwoStep nugget - Model tab

The Model tab contains various graphic views that show summary statistics and distributions for fields of clusters. You can export the data from the model, or you can export the view as a graphic.

Managing IBM Db2 for z/OS Models

Db2® for z/OS® models are added to the canvas and the Models palette in the same way as other IBM® SPSS® Modeler models, and can be used in much the same way.

To score the data directly in Db2 for z/OS, do the following steps:

1. Install SPSS Scoring Adapter in the Db2 for z/OS database where the data is located.
2. Ensure that the stream connects to the Db2 for z/OS database where the data is located.

- [Scoring IBM Db2 for z/OS Models](#)
- [IBM Db2 for z/OS Decision Tree Model Nuggets](#)
- [IBM Db2 for z/OS K-Means model nugget](#)
- [IBM Db2 for z/OS Naïve Bayes model nuggets](#)
- [IBM Db2 for z/OS Regression Tree model nuggets](#)
- [IBM Db2 for z/OS TwoStep model nugget](#)

Scoring IBM® Db2® for z/OS® Models

Models are represented on the canvas by a gold model nugget icon. The main purpose of a nugget is for scoring data to generate predictions, or to allow further analysis of the model properties. Scores are added in the form of one or more extra data fields that can be made visible by attaching a Table node to the nugget and running that branch of the stream, as described later in this section. Some nugget dialog boxes, such as those for Decision Tree or Regression Tree, additionally have a Model tab that provides a visual representation of the model.

The extra fields are distinguished by the prefix \$<id>- added to the name of the target field, where <id> depends on the model, and identifies the type of information being added. The different identifiers are described in the topics for each model nugget.

To view the scores, complete the following steps:

1. Attach a Table node to the model nugget.
2. Open the Table node.
3. Click Run.
4. Scroll to the right of the table output window to view the extra fields and their scores.

Note: The scoring process does not run in the accelerator but in Db2 and consequently requires that the input table for scoring must be physically located in Db2. Therefore, as scoring input, only a Db2-based table or an accelerated table can be used. If the stream uses an accelerator-only table, the following error occurs: "THE STATEMENT CANNOT BE EXECUTED BY DB2 OR IN THE ACCELERATOR."

IBM® Db2 for z/OS Decision Tree Model Nuggets

The Decision Tree model nugget displays the output from the modeling operation and also enables you to set some options for scoring the model.

When you run a stream that contains a Decision Tree model nugget, the node adds two new fields, the names of which are derived from the target.

Table 1. Model-scoring field for Decision Tree

Name of Added Field	Meaning
\$I-target_name	Predicted value for current record.
\$IP-target_name	Confidence value (from 0.0 to 1.0) for the prediction.

Note: Due to limitations in Db2® for z/OS®, the column names might be truncated.

- [IBM Db2 for z/OS Decision Tree Nugget - Model Tab](#)
- [IBM Db2 for z/OS Decision Tree Nugget - Viewer Tab](#)

IBM® Db2® for z/OS® Decision Tree Nugget - Model Tab

The Model tab shows the Predictor Importance of the decision tree model in graphical format. The length of the bar represents the importance of the predictor.

Related information

- [Scoring IBM Db2 for z/OS Models](#)

- [IBM Db2 for z/OS Models - Decision Trees](#)

IBM® Db2® for z/OS® Decision Tree Nugget - Viewer Tab

The Viewer tab shows a tree presentation of the tree model in the same way as the SPSS® Modeler does for its decision tree model.

IBM® Db2 for z/OS K-Means model nugget

K-Means model nuggets contain all of the information captured by the clustering model, as well as information about the training data and the estimation process.

When you run a stream that contains a K-Means model nugget, the node adds two new fields that contain the cluster membership and distance from the assigned cluster center for that record. The new field names are derived from the model name, prefixed by \$KM- for the cluster membership and \$KMD- for the distance from the cluster center. For example, if your model is named Kmeans, the new fields would be named \$KM-Kmeans and \$KMD-Kmeans.

Note: Due to limitations in Db2® for z/OS®, the column names might be truncated.

- [IBM Db2 for z/OS K-Means nugget - Model tab](#)

IBM® Db2® for z/OS® K-Means nugget - Model tab

The Model tab contains various graphic views that show summary statistics and distributions for fields of clusters. You can export the data from the model, or you can export the view as a graphic.

IBM® Db2 for z/OS Naive Bayes model nuggets

When you run a stream that contains a Naive Bayes model nugget, the node adds two new fields, the names of which are derived from the target name.

Table 1. Model-scoring field for Naive Bayes

Name of Added Field	Meaning
\$I-target_name	Predicted value for current record.
\$IP-target_name	Confidence value (from 0.0 to 1.0) for the prediction.

Note: Due to limitations in Db2® for z/OS®, the column names might be truncated.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

IBM® Db2 for z/OS Regression Tree model nuggets

When you run a stream that contains a Regression Tree model nugget, the node adds two new fields, the names of which are derived from the target name.

Table 1. Model-scoring field for Regression Tree

Name of Added Field	Meaning
\$I-target_name	Predicted value for current record.
\$IS-target_name	Estimated standard deviation of the predicted value.

Note: Due to limitations in Db2® for z/OS®, the column names might be truncated.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

- [IBM Db2 for z/OS Regression Tree nugget - Model tab](#)
- [IBM Db2 for z/OS Regression Tree nugget - Viewer tab](#)

IBM® Db2® for z/OS® Regression Tree nugget - Model tab

The Model tab shows the Predictor Importance of the regression tree model in graphical format. The length of the bar represents the importance of the predictor.

IBM® Db2® for z/OS® Regression Tree nugget - Viewer tab

The Viewer tab shows a tree presentation of the tree model in the same way as the SPSS® Modeler does for its regression tree model.

IBM® Db2® for z/OS® TwoStep model nugget

When you run a stream that contains a TwoStep model nugget, the node adds two new fields that contain the cluster membership and distance from the assigned cluster center for that record. The new field names are derived from the model name, prefixed by \$TS- for the cluster membership and \$TSD- for the distance from the cluster center. For example, if your model is named MDL, the new fields would be named \$TS-MDL and \$TSD-MDL.

Architecture and Hardware Recommendations

- [Architecture Description](#)
- [Hardware Recommendations](#)
- [Data Access](#)

Architecture Description

IBM® SPSS® Modeler Server uses a three-tier, distributed architecture. Software operations are shared between the client and the server computers. The advantages of installing and using IBM SPSS Modeler Server (versus the standalone IBM SPSS Modeler), especially when dealing with large data sets, are numerous:

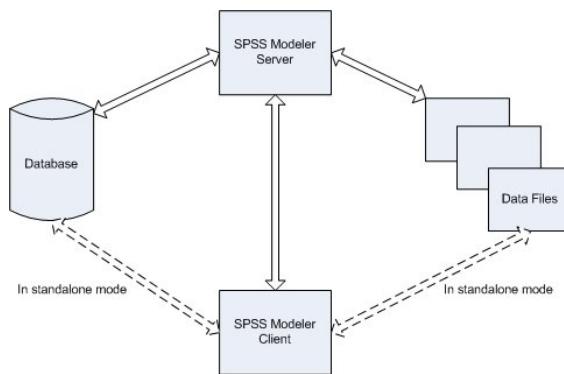
- IBM SPSS Modeler Server can run on UNIX, in addition to Windows, allowing more flexibility in deciding where to install it. On any platform, you can dedicate a faster, larger server computer to data mining processes.
- IBM SPSS Modeler Server is optimized for fast performance. When operations cannot be pushed into the database, IBM SPSS Modeler Server stores the intermediate results as temporary files on disk rather than in RAM. Because servers usually have significant disk space available, IBM SPSS Modeler Server can perform sort, merge, and aggregation operations on very large data sets.
- Using the client-server architecture, you can centralize data-mining processes in your organization. Centralization can help to formalize the role of data mining in your business processes.
- Using administrator tools like the IBM SPSS Modeler Administration Console (included with IBM SPSS Deployment Manager) and IBM SPSS Collaboration and Deployment Services (sold separately), you can monitor data mining processes, ensuring that adequate computing resources are available. With IBM SPSS Collaboration and Deployment Services you can automate certain data mining tasks, manage access to data models, and share results across your organization.

The components of IBM SPSS Modeler's distributed architecture are shown in the "[IBM SPSS Modeler Server Architecture](#)" graphic.

- **IBM SPSS Modeler.** The client software is installed on the end user's computer. It provides the user interface and displays the data mining results. The client is a complete installation of IBM SPSS Modeler software, but when it is connected to IBM SPSS Modeler Server for distributed analysis, its execution engine is inactive. The IBM SPSS Modeler runs on Windows operating systems only.
- **IBM SPSS Modeler Server.** The server software installed on a server computer, with network connectivity to both the IBM SPSS Modeler(s) and the database. IBM SPSS Modeler Server runs as a service (on Windows) or a daemon process (on UNIX), waiting for clients to connect. It handles the execution of streams and scripts created using the IBM SPSS Modeler.
- **Database server.** The database server could be a live data warehouse (for example, Oracle on a large UNIX server) or, to reduce impact on other operational systems, a data mart on a local/departmental server (for example, SQL Server on Windows).

IBM SPSS Modeler Server Architecture

Figure 1. IBM SPSS Modeler Server architecture



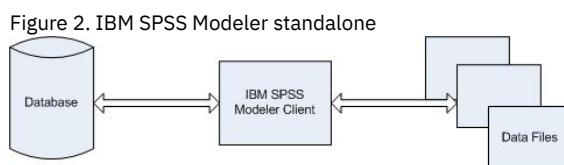
With the distributed architecture, most of the processing occurs on the server computer. When the end user executes a stream, IBM SPSS Modeler sends a description of the stream to the server. The server determines which operations can be executed in SQL and creates the appropriate queries. These queries are executed in the database, and the resulting data are passed to the server for any processing that cannot be expressed using SQL. Once the processing is complete, only the relevant results are passed back to the client.

If necessary, IBM SPSS Modeler Server can execute all IBM SPSS Modeler operations outside of the database. It automatically balances its use of RAM and disk memory to hold data for manipulation. This process makes IBM SPSS Modeler Server fully compatible with flat files.

Load balancing is also available by using a cluster of servers for processing. Clustering is available starting in IBM SPSS Collaboration and Deployment Services 3.5 through the Coordinator of Processes plug-in. See the topic [Load Balancing with Server Clusters](#) for more information. You can connect to a server or cluster managed in the Coordinator of Processes directly through IBM SPSS Modeler's Server Login dialog. See the topic [Connecting to IBM SPSS Modeler Server](#) for more information.

Standalone Client

IBM SPSS Modeler may also be configured to run as a self-contained desktop application, shown in the graphic below. See [IBM SPSS Modeler Support](#) for more information.



Hardware Recommendations

As you plan your IBM® SPSS® Modeler Server installation, you should consider the hardware that you will use. Although IBM SPSS Modeler Server is designed to be speedy, you can maximize its efficiency by using hardware that is sized appropriately for your data mining tasks. Upgrading hardware is often the simplest and most economical way to improve performance across the board.

Dedicated server. Install IBM SPSS Modeler Server on a dedicated server machine where it will not compete for resources with other applications, including any databases to which IBM SPSS Modeler Server may be connecting. Model-building operations in particular are resource-intensive and perform much better when not in competition with other applications.

Note: Although installing IBM SPSS Modeler Server on the same computer as the database can reduce data-transfer time between the database and the server by avoiding network overhead, in most cases the best configuration is to have the server and database on separate machines to avoid competition for resources. Provide a fast connection between the two to minimize the cost of data transfer.

Processors. The number of processors on the machine should be no less than the number of concurrent tasks (simultaneously executing streams) you expect to run on a regular basis. In general, the more processors, the better.

- A single instance of IBM SPSS Modeler Server will accept connections from multiple clients (users), and each client connection can initiate multiple stream executions. One server can therefore have several execution tasks in progress at any one time.
- As a rule of thumb, allow one processor for one or two users, two processors for up to four users, and four processors for up to eight users. Add one additional processor for every two to four users beyond that, depending on the mix of work.
- To the extent that some processing may be pushed back to the database through SQL optimization, it may be possible to share a CPU between two or more users with minimal loss in performance.
- Multithreading capabilities make it possible for a single task to take advantage of multiple processors, so adding processors can improve performance even in cases where only one task is running at a time. Generally, multithreading is used for C5.0 model building and certain data preparation operations (sort, aggregate, and merge). Multithreading is also supported for all nodes that run in IBM SPSS Analytic Server (for example: GLE, Linear-AS, Random Forest, LSVM, Tree-AS, Time Series, TCM, Association Rules, and STP).

64-bit platforms. If you plan to process or build models on very large volumes of data, use a 64-bit machine as your IBM SPSS Modeler Server platform, and maximize the amount of RAM for the machine. For larger data sets, the server can quickly exhaust the per-process memory limits

imposed by 32-bit platforms, forcing data to be spilled to disk and significantly increasing the running time. 64-bit server implementations can take advantage of additional RAM; a minimum of 8 gigabytes (GB) is recommended.

Future needs. Whenever feasible, make sure that server hardware is expandable in terms of memory and CPUs, both to accommodate increases in usage (for example, increased numbers of simultaneous users or increases in the existing users processing requirements) and increased multithreading capabilities of IBM SPSS Modeler Server in the future.

- [Temporary Disk Space and RAM Requirements](#)
-

Temporary Disk Space and RAM Requirements

IBM® SPSS® Modeler Server uses temporary disk space to process large volumes of data. The amount of temporary space that you need depends on the volume and type of data that you process and the type of operations you perform. The data volume is proportional to both the number of rows and the number of columns. The more rows and columns that you process, the more disk space you need.

- [Conditions That Require Temporary Disk Space](#)
 - [Calculating the Amount of Temporary Disk Space](#)
 - [RAM Requirements](#)
-

Conditions That Require Temporary Disk Space

The powerful SQL optimization feature of IBM® SPSS® Modeler Server means that processing can occur in the database (rather than on the server) whenever possible. However, when any of the following conditions are true, SQL optimization cannot be used:

- The data to be processed are held in a flat file rather than in a database.
- SQL optimization is turned off.
- The processing operation cannot be optimized using SQL.

When SQL optimization cannot be used, the following data manipulation nodes and CLEM functions create temporary disk copies of some or all of the data. If the streams used at your site contain these processing commands or functions, you may need to set aside additional disk space on your server.

- Aggregate node
- Distinct node
- Binning node
- Merge node when using the merge-by-key option
- Any modeling node
- Sort node
- Table output node
- @OFFSET functions in which the lookup condition uses @THIS.
- Any @ function, such as @MIN, @MAX, and @AVE, in which the offset parameter is calculated.

Related information

- [Calculating the Amount of Temporary Disk Space](#)
 - [RAM Requirements](#)
-

Calculating the Amount of Temporary Disk Space

In general, IBM® SPSS® Modeler Server needs to be able to write a temporary file that is at least *three times as large* as the original data set. For example, if the data file is 2GB and SQL generation is not used, IBM SPSS Modeler Server will require 6GB of disk space to process the data. Because each concurrent user account creates its own temporary files, you will need to increase the disk space accordingly for each concurrent user.

If you find that your site frequently uses large temporary files, consider using a separate file system for IBM SPSS Modeler's temporary files, created on a separate disk. For best results, a RAID 0 or striped data set that spans multiple physical disks can be used to speed up disk operations, ideally with each disk in the striped file system on a separate disk controller.

Related information

- [Conditions That Require Temporary Disk Space](#)

- [RAM Requirements](#)
-

RAM Requirements

For most processing that cannot be performed in the database, IBM® SPSS® Modeler Server stores the intermediate results as temporary files on disk rather than in memory (RAM). However, for modeling nodes, RAM is used if possible. The Neural Net, Kohonen, and K-Means nodes require large amounts of RAM. If these nodes are frequently used at your site, consider installing more RAM on the server.

In general, the number of bytes of RAM needed can be estimated by

```
(number_of_records * number_of_cells_per_record) * number_of_bytes_per_cell
```

where `number_of_cells_per_record` can become very large when there are nominal fields.

Refer to the system requirements section of the server installation guide for current RAM recommendations. For four or more simultaneous users, even more RAM is recommended. Memory must be shared between concurrent tasks, so scale up accordingly. In general, adding memory is likely to be one of the most cost-effective ways to improve performance across the board.

Related information

- [Conditions That Require Temporary Disk Space](#)
 - [Calculating the Amount of Temporary Disk Space](#)
-

Data Access

To read or write to a database, you must have an ODBC data source installed and configured for the relevant database, with read or write permissions as needed. The IBM SPSS Data Access Pack includes a set of ODBC drivers that can be used for this purpose, and these drivers are available from the download site. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

Supported ODBC drivers

For the latest information on which databases and ODBC drivers are supported and tested for use with IBM® SPSS® Modeler, see the product compatibility matrices on the corporate Support site (<http://www.ibm.com/support>).

Where to install drivers

Note: ODBC drivers must be installed and configured on each computer where processing may occur.

- If you are running IBM SPSS Modeler in local (standalone) mode, the drivers must be installed on the local computer.
- If you are running IBM SPSS Modeler in distributed mode against a remote IBM SPSS Modeler Server, the ODBC drivers need to be installed on the computer where IBM SPSS Modeler Server is installed. For IBM SPSS Modeler Server on UNIX systems, see also "Configuring ODBC drivers on UNIX systems" later in this section.
- If you need to access the same data sources from both IBM SPSS Modeler and IBM SPSS Modeler Server, the ODBC drivers must be installed on both computers.
- If you are running IBM SPSS Modeler over Terminal Services, the ODBC drivers need to be installed on the Terminal Services server on which you have IBM SPSS Modeler installed.

Configuring ODBC drivers on UNIX systems

By default, the DataDirect Driver Manager is not configured for IBM SPSS Modeler Server on UNIX systems. To configure UNIX to load the DataDirect Driver Manager, enter the following commands:

```
cd <modeler_server_install_directory>/bin  
rm -f libspssodbc.so
```

Then run this command if you want to use the UTF8 driver wrapper:

```
ln -s libspssodbc_datadirect.so libspssodbc.so
```

Or run this command instead if you want to use the UTF16 driver wrapper:

```
ln -s libspssodbc_datadirect_utf16.so libspssodbc.so
```

Doing so removes the default link and creates a link to the DataDirect Driver Manager.

Note: The UTF16 driver wrapper is required to use SAP HANA or IBM Db2 CLI drivers for some databases. DashDB requires the IBM Db2 CLI driver.

To configure SPSS Modeler Server:

1. Configure the SPSS Modeler Server start up script modelersrv.sh to source the IBM SPSS Data Access Pack odbc.sh environment file by adding the following line to modelersrv.sh:

```
. /<pathtoSDAPinstall>/odbc.sh
```

Where <pathtoSDAPinstall> is the full path to your IBM SPSS Data Access Pack installation.

2. Restart SPSS Modeler Server.

In addition, for SAP HANA and IBM Db2 only, add the following parameter definition to the DSN in your odbc.ini file to avoid buffer overflows during connection:

DriverUnicodeType=1

Note: The libspssodbc_datadirect_utf16.so wrapper is also compatible with the other SPSS Modeler Server supported ODBC drivers.

Note: The above rules apply specifically to accessing data in a database. Other types of file operations, such as opening and saving streams, projects, models, nodes, PMML, output, and script files, are always done on the client and are always specified in terms of the file system of the client computer. In addition, the Set Directory command in SPSS Modeler sets the working directory for *local* client objects (for example, streams) but does not affect the server's working directory.

UNIX and SPSS Statistics

For information about how to configure SPSS Modeler Server on UNIX to work with the IBM SPSS Statistics data access technology, see [Configuring UNIX Startup Scripts](#).

- [Referencing Data Files](#)
 - [Importing IBM SPSS Statistics Data Files](#)
-

Referencing Data Files

Windows. If you store data on the same computer as IBM® SPSS® Modeler Server, we recommend that you give the path to the data from the perspective of the server computer (for example, C:\ServerData\Sales 1998.csv). Performance is faster when the network is not used to locate the file.

If the data is stored on a different host, we recommend using UNC file references (for example, \\mydataserver\ServerData\Sales 1998.csv). Note that UNC names work only when the path contains the name of a shared network resource. The referencing computer must have permission to read the specified file. If you switch frequently from distributed to local analysis mode, use UNC file references because they work regardless of the mode.

UNIX. To reference data files that reside on a UNIX server, use the full file specification and forward slashes (for example, /public/data/ServerData/Sales 1998.csv). Avoid using the backslash character in the UNIX directory and in filenames for data used with IBM SPSS Modeler Server. It does not matter whether a text file uses UNIX or DOS format—both are handled automatically.

Related information

- [Importing IBM SPSS Statistics Data Files](#)
-

Importing IBM SPSS Statistics Data Files

If you are also running IBM® SPSS® Statistics Server at your site, users may want to import or export IBM SPSS Statistics data while in distributed mode. Recall that when the IBM SPSS Modeler runs in distributed mode, it presents the server's file system. The IBM SPSS Statistics client works in the same way. For importing and exporting to take place between the two applications, both clients must be operating in the same mode. If they are not, their views of the file systems will be different and they will not be able to share files. The IBM SPSS Statistics nodes in IBM SPSS Modeler can automatically start the IBM SPSS Statistics client, but users must first ensure that the IBM SPSS Statistics client is operating in the same mode as IBM SPSS Modeler.

IBM SPSS Modeler Support

This section is intended for administrators and help-desk personnel who support users of IBM® SPSS® Modeler. It covers the following topics:

- How to log on to IBM SPSS Modeler Server (or run standalone by disconnecting from a Server)
 - Data and file systems that users may need
 - User accounts and file permissions pertaining to IBM SPSS Modeler Server
 - Differences in results that users may see when switching between IBM SPSS Modeler Server and IBM SPSS Modeler
- [Connecting to IBM SPSS Modeler Server](#)
 - [Data and File Systems](#)
 - [User Authentication](#)
 - [Differences in Results](#)
-

Connecting to IBM SPSS Modeler Server

IBM® SPSS® Modeler can be run as a standalone application, or as a client connected to IBM SPSS Modeler Server directly or to an IBM SPSS Modeler Server or server cluster through the Coordinator of Processes plug-in from IBM SPSS Collaboration and Deployment Services. The current connection status is displayed at the bottom left of the IBM SPSS Modeler window.

Whenever you want to connect to a server, you can manually enter the server name to which you want to connect or select a name that you have previously defined. However, if you have IBM SPSS Collaboration and Deployment Services, you can search through a list of servers or server clusters from the Server Login dialog box. The ability to browse through the Statistics services running on a network is made available through the Coordinator of Processes.

To Connect to a Server

1. On the Tools menu, click Server Login. The Server Login dialog box opens. Alternatively, double-click the connection status area of the IBM SPSS Modeler window.
2. Using the dialog box, specify options to connect to the local server computer or select a connection from the table.
 - Click Add or Edit to add or edit a connection. See the topic [Adding and Editing the IBM SPSS Modeler Server Connection](#) for more information.
 - Click Search to access a server or server cluster in the Coordinator of Processes. See the topic [Searching for Servers in IBM SPSS Collaboration and Deployment Services](#) for more information.

Server table. This table contains the set of defined server connections. The table displays the default connection, server name, description, and port number. You can manually add a new connection, as well as select or search for an existing connection. To set a particular server as the default connection, select the check box in the Default column in the table for the connection.

Default data path. Specify a path used for data on the server computer. Click the ellipsis button (...) to browse to the required location.

Set Credentials. Leave this box unchecked to enable the **single sign-on** feature, which attempts to log you in to the server using your local computer username and password details. If single sign-on is not possible, or if you check this box to disable single sign-on (for example, to log in to an administrator account), the following fields are enabled for you to enter your credentials.

User ID. Enter the user name with which to log on to the server.

Password. Enter the password associated with the specified user name.

Domain. Specify the domain used to log on to the server. A domain name is required only when the server computer is in a different Windows domain than the client computer.

3. Click OK to complete the connection.

To Disconnect from a Server

1. On the Tools menu, click Server Login. The Server Login dialog box opens. Alternatively, double-click the connection status area of the IBM SPSS Modeler window.
 2. In the dialog box, select the Local Server and click OK.
- [Adding and Editing the IBM SPSS Modeler Server Connection](#)
 - [Searching for Servers in IBM SPSS Collaboration and Deployment Services](#)
 - [Configuring single sign-on](#)
 - [Adding and Editing the IBM SPSS Modeler Server Connection](#)
 - [Searching for Servers in IBM SPSS Collaboration and Deployment Services](#)

Related information

- [Starting IBM SPSS Modeler](#)

Configuring single sign-on

You can connect to an IBM® SPSS® Modeler Server that is running on any supported platform using Single Sign-On. To connect using Single Sign-On, you must first configure your IBM SPSS Modeler server and client machines.

If you are using Single Sign-On to connect to both IBM SPSS Modeler Server and IBM SPSS Collaboration and Deployment Services, you must connect to IBM SPSS Collaboration and Deployment Services before you connect to IBM SPSS Modeler.

IBM SPSS Modeler Server uses Kerberos for Single Sign-On.

Kerberos is a core component of Windows Active Directory, and the following information assumes an Active Directory infrastructure. In particular:

- The client computer is a Windows computer that is joined to an Active Directory domain
- The client user has logged in to the computer using a domain account. The mechanism used to log in is unimportant and may employ a smart card, fingerprint, etc.
- IBM SPSS Modeler Server can validate the client user's credentials by reference to the Active Directory domain controller

This documentation describes how both Windows and UNIX servers can be configured to authenticate this way. Other configurations may be possible but are untested.

To inter-operate with most modern, secure Active Directory installations, you must install the high-strength encryption pack for Java because the required encryption algorithms are not supported by default. You must install the pack for both client and server. An error message such as **Illegal key size** is displayed on the client when a server connection fails because the pack is not installed. See [Installing unlimited strength encryption](#).

- [The Service Principal Name](#)
- [Configuring IBM SPSS Modeler Server on Windows](#)
- [Configuring IBM SPSS Modeler Server on UNIX and Linux](#)
- [Configuring IBM SPSS Modeler client](#)
- [Getting the SSO user's group membership](#)
- [Single sign-on for data sources](#)

The Service Principal Name

Each server instance must register a unique *service principal name (SPN)* to identify itself, and the client must specify the same SPN when it connects to the server.

An SPN for an instance of SPSS® Modeler Server has the form:

```
modelerserver/<host>:<port>
```

For example:

```
modelerserver/jdoemachine.spss.com:28054
```

Note that the host name must be qualified with its DNS domain (**spss.com** in this example), and the domain must map to the Kerberos realm.

The combination of host name and port number makes the SPN unique (because each instance on a given host must listen on a different port). And both client and server already have the host name and port number and so can construct the appropriate SPN for the instance. The additional configuration step required is to register the SPN in the Kerberos database.

Registering the SPN on Windows

If you are using Active Directory as your Kerberos implementation, use the **setspn** command to register the SPN. To run this command, the following conditions must be satisfied:

- You must be logged on to a domain controller
- You must run the command prompt with elevated privileges (run as administrator)
- You must be a member of the Domain Admins group (or have had the appropriate permission delegated to you by a domain administrator)

For more information, refer to the following articles:

- [Setspn Command-Line Reference](#)
- [Delegating Authority to Modify SPNs](#)

For the default instance, listening on the standard port (28054, for example) and running under the Local System account, you must register the SPN against the server computer name. For example:

```
setspn -s modelerserver/jdoemachine.spss.com:28054 jdoemachine
```

For each subsequent (profile) instance, listening on a custom port (for example, 29000) and running under an arbitrary user account (for example, `jdoe`) with the option `start_process_as_login_user` set to `Y`, you must register the SPN against the service user account name:

```
setspn -s modelerserver/jdoemachine.spss.com:29000 jdoe
```

Note that in this case (when the service account is other than Local System), registering the SPN is not sufficient to enable a client to connect. Additional configuration steps are described in the next section.

To see which SPNs are registered to the account `jdoe`:

```
setspn -l jdoe
```

Registering the SPN on UNIX

If you are using Active Directory as your Kerberos implementation, you can use the `setspn` command as described in the previous Windows section; this assumes you have already created the computer or user account in the directory. Or you can use `ktpass`, as illustrated in [Configuring IBM SPSS Modeler Server on UNIX and Linux](#).

If you are using some other Kerberos implementation, then use the Kerberos administration tool to add the service principal to the Kerberos database. To convert the SPN to a Kerberos principal you must append the name of the Kerberos realm. For example:

```
modelerserver/jdoemachine.spss.com:28054@MODELERSO.COM
```

Add this same principal and password to the server's keytab. The keytab must contain an entry for every instance running on the host.

Related information

- [Configuring single sign on](#)
- [Configuring IBM SPSS Modeler Server on Windows](#)
- [Getting the SSO user's group membership](#)

Configuring IBM® SPSS Modeler Server on Windows

In the default scenario where the SPSS® Modeler Server service runs under the Local System account, it uses native Windows APIs to authenticate the user's credentials and no additional configuration is required on the server.

In the alternative scenario where the SPSS Modeler Server service runs under a dedicated user account and `start_process_as_login_user` is set to `Y`, then it uses Java APIs to authenticate the user's credentials and additional configuration is required on the server.

First, verify that the default scenario works. The client should be able to use SSO to connect to the default instance running under the Local System account. This will validate the client-side configuration (that is unchanged). You will need to register the SPN for the default instance as described earlier.

Then perform the following steps:

1. Create the directory `<MODELERSERVER>\config\sso`.
2. Create a file called `krb5.conf` in the `sso` folder you created in step 1. For instructions on how to create this file, see step 3 under [Configuring IBM SPSS Modeler client](#). The file must be the same on the server and client.
3. Use the following command to create the file `krb5.keytab` in the server SSO directory:

```
<MODELERSERVER>\jre\bin\ktab -a <spn>@<realm> -k krb5.keytab
```

For example:

```
"..\jre\bin\ktab.exe" -a modelerserver/jdoemachine.spss.com:29000@SPSS.COM  
-k krb5.keytab
```

This will prompt you for a password. The password you enter must be the password of the service account. So if the service account is `jdoe`, for example, you must enter the password for the user `jdoe`.

The service account itself is not mentioned in the keytab, but earlier you registered the SPN to that account using `setspn`. This means that the password for the service principal and the password for the service account are one and the same.

For each new instance (profile) you create, you must register the SPN for that instance (using `setspn`; see [Configuring server profiles](#) and [The Service Principal Name](#)) and add an entry to the keytab (using `jre\bin\ktab`). There is only one keytab file, and it must contain an entry for every instance that is not running as Local System. The default instance, or any other instance running as Local System, does not need to be in the keytab because it uses Windows APIs to authenticate. Windows APIs do not use the keytab.

To verify that an instance is included in the keytab:

```
ktab.exe -l -e -k krb5.keytab
```

You may see multiple entries for each principal with different encryption types, but this is normal.

Related information

- [The Service Principal Name](#)
- [Configuring single sign-on](#)
- [Getting the SSO user's group membership](#)

Configuring IBM SPSS Modeler Server on UNIX and Linux

Prerequisites

IBM® SPSS® Modeler Server relies on Windows Active Directory (AD) to enable single sign-on, for which the following prerequisites are essential:

- The SPSS Modeler Client (Windows) computer is a member of an Active Directory (AD) domain.
- The client user logs in to the computer using an AD domain account.
- The SPSS Modeler Server (UNIX) computer is identified by a fully-qualified domain name that is rooted in the AD DNS domain. For example, if the DNS domain is `modelersso.com`, then the server hostname might be `myserver.modelersso.com`.
- The AD DNS domain supports both forward and reverse lookups for the SPSS Modeler Server hostname.

If the SPSS Modeler Server machine is not a member of the AD domain you must create a domain user account to represent the service in the directory. For example, you could create a domain account called `ModelerServer`.

To Configure SPSS Modeler Server on UNIX or Linux

1. In the SPSS Modeler Serverconfig folder, create a subfolder called sso.
2. In the sso folder, create a keytab file. The keytab file's generation can be done on the AD side; however, there are different requirements depending on whether the SPSS Modeler Server machine is a member of the AD domain:
 - If the SPSS Modeler Server machine **is** a member of the AD domain, use the computer account name as the service user name:

```
ktpass -princ <spn>@<realm> -mapUser <domain>\<computer account> -pass <password> -out <output file> -ptype KRB5_NT_PRINCIPAL
```

For example:

```
ktpass -princ modelerserver/myserver.modelersso.com:28054@MODELERSO.COM -mapUser MODELERSO\myserver$ -pass Pass1234 -out c:\myserver.keytab -ptype KRB5_NT_PRINCIPAL
```

- If the SPSS Modeler Server machine **is not** a member of the AD domain, specify the domain user account, that you created as a prerequisite, as the service user:

```
ktpass -princ <spn>@<realm> -mapUser <domain>\<user account> -mapOp set -pass <password> -out <output file> -ptype KRB5_NT_PRINCIPAL
```

For example:

```
ktpass -princ modelerserver/myserver.modelersso.com:28054@MODELERSO.COM -mapUser MODELERSO\MyModelerServer -mapOp set -pass Pass1234 -out c:\myserver.keytab -ptype KRB5_NT_PRINCIPAL
```

For more information, see [Ktpass Command-Line Reference](#).

3. Rename the keytab file in the sso folder to krb5.keytab.

Note: If you re-join the server machine to the domain, generate a new keytab file.

4. Create a file called krb5.conf in the sso folder you created in step 1. For instructions on how to create this file, see step 3 under [Configuring IBM SPSS Modeler client](#). The file must be the same on the server and client.

Configuring IBM SPSS Modeler client

1. Enable Java to access the TGT session key:
 - a. From the Start menu, click Run.
 - b. Enter `regedit` and click OK to open the Registry Editor.

- c. Navigate to the registry location appropriate to the operating system of the local machine:
 - On Windows XP: My Computer\HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos
 - On Windows Vista, or Windows 7: My Computer\HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters
 - d. Right-click the folder and select New > DWORD. The name of the new value should be allowtgtsessionkey.
 - e. Set the value of allowtgtsessionkey to a hexadecimal value of 1, that is 0x00000001.
 - f. Close the Registry Editor.
 - g. Note there is a known issue when the user account is a member of the local Administrators group and User Account Control (UAC) is enabled. In this case, the session key in the retrieved service ticket is empty, which causes SSO authentication to fail. To avoid this issue, perform one of the actions:
 - Run the application as Administrator
 - Disable User Account Control
 - Use an account that is not an Administrator account
2. In the config folder of the IBM® SPSS® Modeler installation location, create a folder called sso.
3. In the sso folder, create a krb5.conf file. Instructions for how to create a krb5.conf file can be found at http://web.mit.edu/kerberos/krb5-current/doc/admin/conf_files/krb5_conf.html. An example of a krb5.conf file is provided below:

```
[libdefaults]
    default_realm = MODELERSSO.COM
    dns_lookup_kdc = true
    dns_lookup_realm = true

[realms]
    MODELERSSO.COM = {
        kdc = ad.modelersso.com:88
        admin_server = ad.modelersso.com:749
        default_domain = modelersso.com
    }

[domain_realm]
    .modelersso.com = MODELERSSO.COM
    modelersso.com = MODELERSSO.COM
```

4. Restart the local machine and the server machine.

Getting the SSO user's group membership

When a user logs on to SPSS® Modeler Server using SSO and the server is running non-root, then the name of the authenticated user is not associated with an operating system user account. The server cannot obtain the user's operating system group membership. So how is group configuration performed in this case?

We assume the user is registered in an LDAP directory (which could be Active Directory) and we can request the group membership from the LDAP server. SPSS Modeler Server can query the LDAP provider in IBM® SPSS Collaboration and Deployment Services for the group membership.

There are two properties in options.cfg on the SPSS Modeler Server that control the server's access to the IBM SPSS Collaboration and Deployment Services Repository:

```
repository_enabled, N
repository_url, ""
```

To enable group lookup, you must set both properties. For example:

```
repository_enabled, Y
repository_url, "http://jdoemachine.spss.ibm.com:9083"
```

The repository connection is only used for SSO group lookup, so you do not need to change these property settings unless you need this feature.

For group lookup to work properly, you must configure your repository first to add an LDAP or Active Directory provider and then to enable SSO using that provider:

1. Start IBM SPSS Deployment Manager client and select File > New > Administered Server Connection... to create an administered server connection for your repository (if you do not have one already).
2. Log on to the administered server connection and expand the Configuration folder.
3. Right-click Security Providers, choose New > Security provider definition..., and enter the appropriate values. Click Help in the dialog for more information.
4. Expand the Single Sign-On Providers folder, right-click Kerberos SSO Provider, and select Open.
5. Click Enable, select your security provider, and then click Save. You do not have to fill in any other details here unless you want to use SSO (simply having the provider enabled is sufficient to allow the group lookup).

Important: For group lookup to work properly, the Kerberos provider you configure here must be the same as the provider you configured for SPSS Modeler Server. In particular, they must be working within the same Kerberos realm. So if a user logs on to SPSS Modeler Server using SSO

and it identifies him as `jdoe@SPSS.COM` (where `SPSS.COM` is the realm), it will expect the security provider in IBM SPSS Collaboration and Deployment Services to recognize that user principal name and return the corresponding group membership from the LDAP directory.

Related information

- [The Service Principal Name](#)
 - [Configuring IBM SPSS Modeler Server on Windows](#)
 - [Configuring single sign-on](#)
-

Single sign-on for data sources

You can connect to databases from IBM® SPSS® Modeler using single sign-on. If you want to create a database connection using single sign-on, you must first use your ODBC management software to properly configure a data source and single sign-on token. Then when connecting to a database in IBM SPSS Modeler, IBM SPSS Modeler will use that same single sign-on token and the user will not be prompted to log on to the data source.

However, if the data source was not configured properly for single sign-on, IBM SPSS Modeler will prompt the user to log on to the data source. The user will still be able to access the data source after providing valid credentials.

For complete details about configuring ODBC data sources on your system with single sign-on enabled, see your database vendor documentation. Following is an example of the general steps that may be involved:

1. Configure your database so it can support Kerberos single sign-on.
 2. On the IBM SPSS Modeler Server machine, create an ODBC data source and test it. The DSN connection should not require a user ID and password.
 3. Connect to IBM SPSS Modeler Server using single sign-on and begin using the ODBC data source created and validate in step 2.
-

Adding and Editing the IBM SPSS Modeler Server Connection

You can manually edit or add a server connection in the Server Login dialog box. By clicking Add, you can access an empty Add/Edit Server dialog box in which you can enter server connection details. By selecting an existing connection and clicking Edit in the Server Login dialog box, the Add/Edit Server dialog box opens with the details for that connection so that you can make any changes.

Note: You cannot edit a server connection that was added from IBM® SPSS® Collaboration and Deployment Services, since the name, port, and other details are defined in IBM SPSS Collaboration and Deployment Services. Best practice dictates that the same ports should be used to communicate with both IBM SPSS Collaboration and Deployment Services and SPSS Modeler Client. These can be set as `max_server_port` and `min_server_port` in the options.cfg file.

To Add Server Connections

1. On the Tools menu, click Server Login. The Server Login dialog box opens.
 2. In this dialog box, click Add. The Server Login Add/Edit Server dialog box opens.
 3. Enter the server connection details and click OK to save the connection and return to the Server Login dialog box.
- **Server.** Specify an available server or select one from the list. The server computer can be identified by an alphanumeric name (for example, `myserver`) or an IP address assigned to the server computer (for example, `202.123.456.78`).
 - **Port.** Give the port number on which the server is listening. If the default does not work, ask your system administrator for the correct port number.
 - **Description.** Enter an optional description for this server connection.
 - **Ensure secure connection (use SSL).** Specifies whether an SSL (**Secure Sockets Layer**) connection should be used. SSL is a commonly used protocol for securing data sent over a network. To use this feature, SSL must be enabled on the server hosting IBM SPSS Modeler Server. If necessary, contact your local administrator for details.

To Edit Server Connections

1. On the Tools menu, click Server Login. The Server Login dialog box opens.
2. In this dialog box, select the connection you want to edit and then click Edit. The Server Login Add/Edit Server dialog box opens.
3. Change the server connection details and click OK to save the changes and return to the Server Login dialog box.

Related information

- [Starting IBM SPSS Modeler](#)

Searching for Servers in IBM SPSS Collaboration and Deployment Services

Instead of entering a server connection manually, you can select a server or server cluster available on the network through the Coordinator of Processes, available in IBM® SPSS® Collaboration and Deployment Services. A server cluster is a group of servers from which the Coordinator of Processes determines the server best suited to respond to a processing request.

Although you can manually add servers in the Server Login dialog box, searching for available servers lets you connect to servers without requiring that you know the correct server name and port number. This information is automatically provided. However, you still need the correct logon information, such as username, domain, and password.

Note: If you do not have access to the Coordinator of Processes capability, you can still manually enter the server name to which you want to connect or select a name that you have previously defined. See the topic [Adding and Editing the IBM SPSS Modeler Server Connection](#) for more information.

To search for servers and clusters

1. On the Tools menu, click Server Login. The Server Login dialog box opens.
2. In this dialog box, click Search to open the Search for Servers dialog box. If you are not logged on to IBM SPSS Collaboration and Deployment Services when you attempt to browse the Coordinator of Processes, you will be prompted to do so.
3. Select the server or server cluster from the list.
4. Click OK to close the dialog box and add this connection to the table in the Server Login dialog box.

Related information

- [Starting IBM SPSS Modeler](#)

Data and File Systems

Users working with IBM® SPSS® Modeler Server will probably need to access data files and other data sources on the network, as well as save files on the network. They may need the following information, as applicable:

- **ODBC data source information.** If users need access to ODBC data sources defined on the server computer, they will need the names, descriptions, and login information (including database login IDs and passwords) for the data sources.
- **Data file access.** If users need to access data files on the server computer or elsewhere on the network, they will need the names and locations of the data files.
- **Location for saved files.** When users save data while connected to IBM SPSS Modeler Server, they may attempt to save files on the server computer. However, this is often a write-protected location. If so, let users know where they should save data files. (Typically, the location is the user's home directory.)

User Authentication

IBM® SPSS® Modeler Server uses the operating system on the server machine to authenticate users who connect to the server. When a user connects to SPSS Modeler Server, all operations that are performed on behalf of the user are performed in the user's security context. Access to database tables is subject to user and/or password privileges in the database itself.

Windows. On Windows, any user with a valid account on the host network can log on. With the default authentication, users must have modify access rights to the `<modeler_server_install>|Tmp` directory. Without these rights, users cannot log on to SPSS Modeler Server from the client using the default authentication on Windows.

UNIX. By default, SPSS Modeler Server is assumed to run as root on UNIX. This allows any user with a valid account on the host network to log on and limits users' file access to their own files and directories. However, you can configure SPSS Modeler Server to run without root privilege. In this case, you must create a private password database to be used for authentication, and all SPSS Modeler users must share a single UNIX user account (and, consequently, share access to data files). For more information, see [Configuring as non-root using a private password database](#).

Configuring PAM

On the Linux platform, SPSS Modeler Server uses the Pluggable Authentication Module (PAM) for authentication.

To use PAM authentication the appropriate PAM modules must be correctly configured on the host system; for example, for PAM to interface with LDAP, a PAM LDAP module must exist on the host OS and be correctly configured. Refer to the operating system documentation for further

information. This is a prerequisite for SPSS Modeler Server to be able to use PAM.

To configure SPSS Modeler Server to use PAM, edit the SPSS Modeler Server "options.cfg" file and add (or edit) the line **authentication_methods, pam**.

You can use the service name **modelerserver** to provide a specific PAM configuration for SPSS Modeler Server if required. For example, the following steps explain how to configure for Red Hat Linux:

1. Change to the PAM configuration directory. For example: **/etc/pam.d**.
2. Using a text editor, create a new file called "modelerserver".
3. Add the PAM configuration information that you want to use. For example:

```
auth    include    system-auth
account    include    system-auth
password    required    pam_deny.so
session    required    pam_deny.so
```

Note: These lines might vary depending on your particular configuration. For more information, see the Linux documentation.

4. Save the file and restart the Modeler service.

- [Permissions](#)
- [File Creation](#)

Permissions

Windows. A user connecting to server software that is installed on an *NTFS* drive must login with an account that has the following permissions.

- Read and execute permissions to the server's installation directory and its subdirectories
- Read, execute, and write permissions to the directory location for temporary files.

In Windows Server 2008 and later, you cannot assume that users have these permission. Be sure to explicitly set permissions as needed.

If the server software is installed on a *FAT* drive, you do not need to set permissions because all files allow users to have full control.

UNIX. If you are not using internal authentication, a user connecting to the server software must login with an account that has the following permissions:

- Read and execute permissions to the server's installation directory and its subdirectories
- Read, execute, and write permissions to the directory location for temporary files.

File Creation

When IBM® SPSS® Modeler Server accesses and processes data, it often has to keep a temporary copy of that data on disk. The amount of disk space that will be used for temporary files depends on the size of the data file that the end user is analyzing and the type of analysis that he or she is performing. See the topic [Temporary Disk Space and RAM Requirements](#) for more information.

UNIX. The UNIX versions of IBM SPSS Modeler Server use the UNIX **umask** command to set file permissions for the temporary files. You can override the server's default permissions. See the topic [Controlling Permissions on File Creation](#) for more information.

Differences in Results

Users who run analyses in both modes may see slight differences in the results between IBM® SPSS® Modeler and IBM SPSS Modeler Server. The discrepancy usually occurs because of record ordering or rounding differences.

Record ordering. Unless a stream explicitly orders records by sorting them, the order in which records are presented may vary between streams executed locally and those executed on the server. There may also be differences in order between operations run within a database and those run in IBM SPSS Modeler Server. These differences are due to the different algorithms used by each system to implement functions that may reorder records, such as aggregation. Also, note that SQL does not specify the order in which records are returned from a database in cases where there is no explicit ordering operation.

Rounding differences. IBM SPSS Modeler running in local mode uses a different internal format for storing floating point values than does IBM SPSS Modeler Server. Due to rounding differences, results might vary slightly between each version.

IBM® SPSS® Modeler Administration

- [Starting and Stopping IBM SPSS Modeler Server](#)
 - [Handling Unresponsive Server Processes \(UNIX Systems\)](#)
 - [Configuring server profiles](#)
 - [Administration](#)
 - [IBM SPSS Modeler Server Administration](#)
 - [Using SSL to secure data transfer](#)
 - [Configuring groups](#)
 - [Log files](#)
-

Starting and Stopping IBM SPSS Modeler Server

IBM® SPSS® Modeler Server runs as a service on Windows or as a daemon process on UNIX.

Scheduling note: Stopping IBM SPSS Modeler Server disconnects end users and terminates their sessions, so try to schedule server restarts during periods of low usage. If this is not possible, be sure to notify users before stopping the server.

- [To Start, Stop, and Check Status on Windows](#)
 - [To Start, Stop, and Check Status on UNIX](#)
-

To Start, Stop, and Check Status on Windows

On Windows, you control IBM® SPSS® Modeler Server with the Services dialog box in the Windows Control Panel.

1. **Windows XP.** Open the Windows Start menu. Choose Settings and then Control Panel. Double-click Administrative Tools and then Services.
2. **Windows 2003 or 2008.** Open the Windows Start menu. Choose Control Panel, then Administrative Tools, then Services.
3. Select the IBM SPSS Modeler Server <nn.n> service. You can now check its status, start or stop it, and edit startup parameters, as appropriate.

By default, the service is configured for automatic startup, which means that if you stop it, it will restart automatically when the computer is rebooted. When started this way, the service runs unattended, and the server computer can be logged off without affecting it.

To Start, Stop, and Check Status on UNIX

On UNIX, you start or stop IBM® SPSS® Modeler Server by running the *modelersrv.sh* script in the IBM SPSS Modeler Server installation directory.

1. Change to the IBM SPSS Modeler Server installation directory. For example, at a UNIX command prompt, type

```
cd /usr/modelersrv
```

where *modelersrv* is the IBM SPSS Modeler Server installation directory.

2. To start the server, at the command prompt, type

```
./modelersrv.sh start
```

3. To stop the server, at the command prompt, type

```
./modelersrv.sh stop
```

4. To check the status of IBM SPSS Modeler Server, at a UNIX command prompt, type

```
./modelersrv.sh list
```

and look at the output, which is similar to what the UNIX *ps* command produces. The first process in the list is the IBM SPSS Modeler Server daemon process, and remaining processes are IBM SPSS Modeler sessions.

The IBM SPSS Modeler Server installation program includes a script (*auto.sh*) that configures your system to start the server daemon automatically at boot time. If you have run that script and then stop the server, the server daemon will restart automatically when the computer is rebooted. See the topic [Automatically Starting and Stopping IBM SPSS Modeler Server](#) for more information.

UNIX kernel limits

You must ensure that kernel limits on the system are sufficient for the operation of IBM SPSS Modeler Server. The data, memory, file, and processes ulimits are particularly important and should be set to unlimited within the IBM SPSS Modeler Server environment. To do this:

1. Add the following commands to modelersrv.sh:

```
ulimit -d unlimited  
ulimit -m unlimited  
ulimit -f unlimited  
ulimit -u unlimited
```

In addition, set the stack limit to the maximum allowed by your system (`ulimit -s xxxx`), for example:

```
ulimit -s 64000
```

2. Restart IBM SPSS Modeler Server.

Handling Unresponsive Server Processes (UNIX Systems)

IBM® SPSS® Modeler Server processes may become unresponsive for several reasons, including situations where they make a system call or ODBC driver call that becomes blocked (call never returns, or takes a very long time to return). When UNIX processes enter this state, they can be cleaned up using the UNIX `kill` command (interrupts initiated by IBM SPSS Modeler client, or the closing of IBM SPSS Modeler client, will have no effect). A `kill` command is provided as an alternative to the normal `stop` command, and enables an administrator to use `modelersrv.sh` to easily issue the appropriate `kill` command.

On systems which are susceptible to the accumulation of unusable (“zombie”) server processes, we recommend that IBM SPSS Modeler Server is stopped and restarted at regular intervals, using the following sequence of commands:

```
cd modeler_server_install_directory  
. ./modelersrv.sh stop  
. ./modelersrv.sh kill
```

Those IBM SPSS Modeler processes that are ended using the `modelersrv.sh kill` command will leave behind temporary files (from the temporary directory) that will need to be removed manually. Temporary files may be left behind in some other situations too, including application crashes due to resource exhaustion, user interrupts, system crashes, or other reasons. Therefore we recommend that, as part of the process of restarting IBM SPSS Modeler Server at regular intervals, all remaining files are removed from the IBM SPSS Modeler temporary directory.

Once all server processes have been closed and temporary files have been removed, IBM SPSS Modeler Server can be safely restarted.

Configuring server profiles

Server profiles allow you to run multiple, independent instances of SPSS® Modeler Server from a single installation. To a client, they will appear to be separate servers located on the same host but listening on different port numbers. Having multiple instances sharing one installation benefits administrators because it simplifies maintenance. Subsequent instances after the first can be created and deleted more quickly than would be required for a full installation and uninstallation, and Fix Packs only need be applied once.

The reason for running multiple server instances on the same host is to be able to configure each instance separately. If all the instances are identical, there is nothing to be gained. In particular, if the instances run *non-root* (so that all sessions share the same user account), each instance can use a different user account to provide data isolation between user groups. For example, a user logging in to an instance **A** will be allocated a session owned by some particular **User-A** and will have access only to that user's files and folders, whereas a user logging in to instance **B** will see a different set of files and folders accessible to **User-B**. This can be used in conjunction with group configuration so that logging on to a particular instance is restricted to specific groups, meaning that end users can only log on to the instance (or instances) appropriate to their role. See [Configuring groups](#).

In a standard SPSS Modeler Server installation, the folders config, data, and tmp are specific to a server instance. The purpose of the config folder is for the instance to have a private configuration, and the data and tmp folders support data isolation. Each instance has a private copy of these folders, and everything else is shared.

Note that much of the server configuration can remain common (database settings, for example), so a profile configuration will override the common configuration. The server will look first in the profile configuration and then fall back to the default. The files that are most likely to be changed for a profile are options, groups, and passwords.

See [Profile structure](#) for more information.

For information on how to configure a profile to use SSO, see [Configuring single sign-on](#). This requires you to register a Service Principal Name (SPN), perform some configuration if the Windows Service account is not local, and in some cases enable group lookup.

- [Working with server profiles](#)
- [Profile structure](#)
- [Profile scripts](#)

Related information

- [Working in the enterprise](#)
- [Configuring single sign on](#)
- [Configuring groups](#)

Working with server profiles

Following are some common use cases for server profiles. Some of these uses are supported via the use of scripts (see [Profile scripts](#)) and may require administrative/root privileges.

Creating a server profile

An SPSS® Modeler Server administrator named Jane uses a script to create a new profile:

- Jane must specify a unique name for the profile (it cannot be an existing profile name). If the profiles directory does not already exist, it is created for Jane. Then a new sub-directory is created in the profiles directory with name Jane specified, containing the directories config, data, log, and tmp.
- If Jane chooses, she can also specify the name of an existing profile to use as a template, in which case the content of the config folder within the existing profile is copied to the new profile. If she does not specify a template, or if the existing profile does not include an options file even though it should, then an empty options file is created in the new profile.
- Jane may also choose to specify a port number for the profile, in which case the port number is written as the value of the `port_number` property in the profile's options file. If she does not specify a port number, a value is chosen for her and written to the options file.
- Jane can also choose to specify the name of an operating system group that will have exclusive access to the profile in which case group configuration is enabled in the options file. In this case, a groups file is created that denies login to all but the specified group.

Configuring a server profile

Server administrator Jane configures a profile either by manually editing the profile configuration files, or by using the IBM® SPSS Modeler Administration Console in IBM SPSS Deployment Manager to connect to the profile service.

Creating a Windows service for a serverprofile

On Windows, the administrator uses a script to create a service for a specified profile:

- Jane must specify the name of an existing profile, and then a service instance is created for that profile. The command line for the service will include the `profile` argument. The name of the service will follow a standard pattern including the profile name.
- Jane might need to use the service administration console later and edit the service properties if she needs to change the user name and password for the service (when running non-root).

On UNIX, there are also ways to create "services" that start automatically when the system boots. The administrator may want to create profile services using these mechanisms, but note they are not officially supported by IBM SPSS Modeler.

Managing Windows services for server profiles

Administrators can use a script to perform the following tasks:

- See which server profile services are running
- Start a particular service
- Start all services
- Stop a particular service
- Stop all services

When starting or stopping all services, the list of profiles is obtained by searching the sub-directories of the profiles directory.

Deleting a server profile's Windows service

On Windows, administrators can use a script to delete a service for a specified profile (if a service exists for the profile). The name of the profile must be specified.

Removing a server profile

After stopping the profile's service, administrators can remove a profile by deleting its folder from inside the profiles directory.

Updating SPSS Modeler Server

When applying a Fix Pack to SPSS Modeler Server, the Fix Pack is applied to all server profiles. On Windows, all profile services are stopped and restarted automatically. On UNIX, you must manually stop and restart them.

Uninstalling SPSS Modeler Server

When SPSS Modeler Server is uninstalled, all server profiles are uninstalled. Note that the profiles directory and any profiles it contains are not removed automatically. They must be deleted manually. On Windows, all profile services are uninstalled automatically. On UNIX, you must manually remove them.

Installing a new version of SPSS Modeler Server

When installing a new version of SPSS Modeler Server, any existing server profiles are not migrated automatically. An administrator must manually copy profiles from one installation to the next (and edit the configurations where necessary) to recreate the services.

Related information

- [Working in the enterprise](#)
- [Profile structure](#)
- [Profile scripts](#)

Profile structure

The profiles directory

Server profiles are stored in a directory location chosen by the server administrator. The default location is a directory named profiles in the [server install path]\config\ directory on the SPSS® Modeler Server, but we recommend using a different directory for profile storage for the following reasons:

- Profiles can be shared between nodes in a cluster
- Profiles can be preserved across upgrades
- Administrators and other users who configure profiles do not need to be granted write authority to the SPSS Modeler Server installation directory

The profiles directory does not exist after a fresh SPSS Modeler Server installation. It is created when the first profile is created.

The profiles directory contains one sub-directory for each profile, and the sub-directory name matches the profile name. Because the directory name and the profile name are the same, the profile name cannot include characters that are not valid in file names. Nor should profile names contain spaces, because they are likely to cause problems in scripts. Also keep in mind that profile names must be unique within a single installation.

The only way to identify all the profiles for an installation is to identify the sub-directories of the profiles directory. There is no separate list of profiles maintained anywhere. There is also no limit on the number of profiles that can be created for an installation, aside from what can be tolerated by the host system.

In the profiles directory, the sub-directory for any given profile must contain at least one directory called config, and within that directory there must be at least one file called options.cfg that defines the profile configuration. This file contains a subset of the settings in the standard SPSS Modeler Server options.cfg file (located in [server install path]\config), as many as are needed for the profile. Settings not present in the profile configuration must be set from the common options file in the installation config directory. The profile configuration must contain at least a setting for port_number because every profile service must listen on a different port number.

The profile configuration may include other *.cfg files normally found in the installation config directory, in which case these are read instead of the standard files (only the options file is cumulative). Additional files most likely to be included in a profile configuration are groups and passwords. Files that are ignored in a profile configuration include the JVM and SSO configuration files which are shared across all profiles.

A profile directory may also contain data and tmp directories that override the common data and tmp file locations, unless alternative locations are specified in the profile configuration.

If you use profiles to achieve data isolation, be sure permissions are set appropriately on the relevant directories.

The profiles configuration file

The location of the profiles directory is specified in a new configuration file called [server install path]\config\profiles.cfg. This shares a common format with other configuration files in the same directory, and the key for setting the profiles directory is **profiles_directory**. For example:

```
profiles_directory, "C:\\SPSS\\Modeler\\profiles"
```

A separate file is used for profile configuration (rather than adding settings to the standard options file) for two reasons:

- The profile configuration determines how the options files are read, so there is an intrinsic difficulty in defining one in the other
- The profiles configuration file is designed to be managed automatically, using scripts, so in simple cases users need not be concerned with it at all (but it can still be safely hand-edited to support more complex scenarios)

Aside from the location of the profiles directory, the only other entry in profiles.cfg is a port number. For example:

```
profile_port, 28501
```

This is the default port number for the next profile to be created, and it is incremented automatically each time a profile is created using a script. The profiles.cfg file is created only as needed, so it does not exist in a fresh installation.

Starting a profile

The service executable (modelerserver.exe) accepts an additional argument, **profile**, that identifies the profile for the service:

```
modelerserver -server profile=<profile-name>
```

Multiple services can run from the same installation if each service uses a different profile. If the profile argument is omitted, the service uses the common installation defaults without any profile overrides.

When invoked with the **profile** argument, the service:

- Reads [server install path]\config\profiles.cfg to obtain the location of the profiles directory
- Reads [profiles directory]\[profile name]\config\options.cfg to obtain the profile configuration (in particular, the port number)

If either step fails for any reason, the service prints an error message to the log and stops. If the service is invoked with a profile and it cannot load the profile, then it will not run.

Environment variables

The service defines some additional environment variables so that path names, etc., can be expressed without knowledge of the current profile:

Table 1. Environment variables

Variable	Value
PROFILE_NAME	The name of the current profile, or the empty string if no profile has been specified.
MODELERPROFILE	The full path to the directory for the current profile (for example, \$MODELERSERVER\profiles\\$PROFILE_NAME). If no profile has been specified, the value is the same as the \$MODELERSERVER.
MODELERDATA	The full path to the data directory for the current profile (for example, \$MODELERPROFILE\data). If no profile has been specified, the value will point to the standard data directory \$MODELERSERVER\data

These environment variables are set by the service process, so they are only visible within that process and any child processes it creates. If you set these variables outside the service process, they will be ignored and redefined within the process as described.

Logging

Each profile service expects to have a separate, private folder in which to place its log files. There is one copy of server_logging.log, etc., for each profile.

The default **log4cxx.properties** configuration in the installation's config directory uses the PROFILE_NAME environment variable to identify the log directory for the service:

```
log4j.appenders.LoggingAppender.File=${ALLUSERSPROFILE}/IBM/SPSS/ModelerServer/17/log/${PROFILE_NAME}/server_logging.log
```

You can change the log location for all profiles by changing the line above and including one of the two profile-specific environment variables, either PROFILE_NAME or MODELERPROFILE. For example, to relocate the log directory within the profile directory:

```
log4j.appenders.LoggingAppender.File=${MODELERPROFILE}/log  
/server_logging.log
```

Alternatively, you can change the log location for a particular profile by creating and editing a copy of the log4cxx.properties file in the profile configuration.

Related information

- [Working in the enterprise](#)
 - [Working with server profiles](#)
 - [Profile scripts](#)
-

Profile scripts

The scripts described in this section are provided to assist with the creation and management of SPSS® Modeler Server profiles. All of the scripts are included in the scripts/profiles directory of the SPSS Modeler Server installation directory (for example, C:\Program Files\IBM\SPSS\ModelerServer\18\scripts\profiles).

- [Common script \(for all platforms\)](#)
- [Windows scripts](#)
- [UNIX script](#)

Related information

- [Working in the enterprise](#)
 - [Working with server profiles](#)
 - [Profile structure](#)
 - [Common script \(for all platforms\)](#)
 - [Windows scripts](#)
 - [UNIX script](#)
-

Common script (for all platforms)

The following script helps create and manage profiles. Variants of this script are provided with different extensions for different platforms (.bat for Windows and .sh for UNIX). The operation is the same in each case.

Creating a profile

```
create_profile [options] <profile-name>
```

Creates a new profile with the specified name. The profile name must be appropriate for use as a directory name on the server host (because the script will create a directory with that name) and should not contain spaces. The name must be distinct from any existing profile name.

Options:

```
-d, --profiles-directory<profiles-directory>
```

Specifies the profiles directory in which this and all subsequent profiles should be created. You must specify this only for the first profile, but a good practice is to specify it every time. If you omit it the first time, a default location will be chosen. If you change the profiles directory on a subsequent call, the new profile will be created in the new location, but any existing profiles will be ignored unless they are moved separately to the new location.

```
-t, --template <profile-name>
```

Specifies the name of an existing profile to use as a template. The profile configuration is copied from the existing profile to the new profile, and only the port number is changed.

```
-p, --port-number <port-number>
```

Specifies the port number for the profile service. The port number must be unique to this profile. If you omit the port number, a default will be chosen.

```
-g, --group-name <group-name>
```

Specifies the name of an operating system group that will have exclusive access to this profile. The profile is configured to allow login access only to members of this group.

File system permissions are not changed, so you must perform that action separately.

Examples:

```
scripts\profiles\create_profile.bat -d C:\Modeler\Profiles comet
```

Creates a new profile called **comet** in the directory C:\Modeler\Profiles. The profile will listen on a default port number. To determine the port number, open the options.cfg file that is generated for the profile (in this example, C:\Modeler\Profiles\comet\config\options.cfg).

```
scripts\profiles\create_profile.bat --template comet --group-name "Meteor Users"  
--port-number 28510 meteor
```

Creates a new profile called **meteor** in the directory C:\Modeler\Profiles (remembered from the previous command). The profile will listen on port 28510 and login access will be allowed only to members of the group **Meteor Users**. All other configuration options will be copied from the existing profile **comet**.

Related information

- [Working in the enterprise](#)
 - [Profile scripts](#)
 - [Common script \(for all platforms\)](#)
 - [Windows scripts](#)
 - [UNIX script](#)
-

Windows scripts

These scripts assist with the creation and management of Windows services for SPSS® Modeler Server profiles. They use the Windows Service Control program (SC.EXE) to perform the requested operations, and the script output comes from SC.EXE unless otherwise noted. You must have administrator privileges on the local machine to perform most of these tasks.

See [Microsoft's TechNet documentation](#) about SC.EXE for more information.

Creating a Windows service for a profile

```
create_windows_service [options] <profile-name>
```

Creates the Windows service for a specified profile. You must have administrator privileges to create a service. Use the Services Management Console to set additional properties for the service after it is created (for example, to set the account details for the service log on).

Options:

```
-u, --service-user <account-name>
```

Specifies the account used for the service log on (passim). This can be a local user account, a domain user account, or the local computer name (standing for the local system account). The default is the local system account. If you specify an account other than the local system account, you must go to the Services Management Console and set the password for the account before the service will start.

```
-s, --register-spn
```

Registers a Service Principal Name (SPN) for the service so that clients can connect using Kerberos SSO. You must specify the service login account in this case (-u) so that the SPN can be registered to that account. You must have domain administrator privileges to use this option (or have been delegated the authority to register an SPN).

```
-H, --service-host <host-name>
```

Specifies the host name to use in construction of the SPN. This must be the host name clients will connect through, and it must be qualified with a domain name that maps to the Kerberos realm (in a simple active directory configuration, the domain name and the Kerberos realm are one and the same).

Examples:

```
scripts\profiles\create_windows_service.bat comet
```

Creates a Windows service for the **comet** profile. The service is owned by the local system account, and clients are expected to log in with a user name and password.

```
scripts\profiles\create_windows_service.bat -s -H modelerserver.mycompany.com -u  
MYCOMPANY\ProjectMeteor meteor
```

Creates a Windows service for the **meteor** profile. The service is owned by the **ProjectMeteor** domain account, and clients can log in using SSO. The service will not start until you go to the Services Management Console and set the password for the **ProjectMeteor** account. The account will automatically be granted the right to log in as a service.

Deleting a Windows service for a profile

```
delete_windows_service [options] <profile-names...>
```

Deletes the Windows services for the specified profiles. You must have administrator privileges to delete a service.

Options:

```
-s, --summary
```

Lists the names of the services that were deleted. Services that do not exist or cannot be deleted will not be listed. Without this option, the deletion status of all the specified services will be listed.

```
-a, --all
```

Deletes the services for all profiles.

Examples:

```
scripts\profiles\delete_windows_service.bat comet  
Deletes the Windows services for the comet profile.  
scripts\profiles\delete_windows_service.bat --all  
Deletes the Windows services for all profiles.
```

Starting a Windows service for a profile

```
start_windows_service [options] <profile-names...>
```

Starts the Windows services for the specified profiles. You must have administrator privileges to start a service.

Options:

```
-s, --summary  
Lists the names of the services that were started. Services that are already running or cannot be started will not be listed. Without this option, the status of all the listed services is listed.  
-a, --all  
Starts the services for all profiles.
```

Examples:

```
scripts\profiles\start_windows_service.bat -s comet meteor  
Attempts to start the Windows services for the comet and meteor profiles, and lists the names of the services that were successfully started.
```

Stopping a Windows service for a profile

```
stop_windows_service [options] <profile-names...>
```

Stops the Windows services for the specified profiles. You must have administrator privileges to stop a service.

Options:

```
-s, --summary  
Lists the names of the services that were stopped. Services that are already stopped or cannot be stopped will not be listed. Without this option, the status of all the listed services is listed.  
-a, --all  
Stops the services for all profiles.
```

Examples:

```
scripts\profiles\stop_windows_service.bat -a -s  
Attempts to stop the Windows services for all profiles, and prints the names of those that were successfully stopped. The set of all profiles is obtained from the profiles directory.
```

Querying the state of a Windows service for a profile

```
query_windows_service [options] <profile-names...>
```

Shows the status of the Windows services for the specified profiles. You do not need administrator privileges to query a service.

Options:

```
-s, --summary  
Lists just the names of the services and their current state (RUNNING, STOPPED, etc.). If a service cannot be queried for any reason (for example, if it does not exist), the status is reported as UNKNOWN. Without this option, the full status of all the listed services is listed.  
-a, --all  
Queries the service status for all profiles.
```

Examples:

```
scripts\profiles\query_windows_service.bat -a  
Reports the full service status for all profiles.
```

Related information

- [Working in the enterprise](#)
- [Configuring single sign on](#)
- [Configuring groups](#)

UNIX script

The existing UNIX script that manages the SPSS® Modeler Server service now accepts an additional `profile` argument so that SPSS Modeler Server profile services can be managed independently.

`modelersrv.sh [options] {start|stop|kill|list}`

Manages the main SPSS Modeler Server service. See [IBM SPSS Modeler Administration](#) for more information.

Options:

`-p, --profile <profile-name>`

Manages the service instance for the specified profile. When this argument is used, the specified command applies only to the instance for the specified profile. When this argument is absent, the `start` command starts just the default instance (a service with no profile), but the `stop`, `kill`, and `list` commands apply to all active instances.

Examples:

```
./modelersrv.sh --profile comet start  
Starts the service for the comet profile.  
./modelersrv.sh --profile meteor start  
Starts the service for the meteor profile.  
./modelersrv.sh list  
Lists the processes for all active services.  
./modelersrv.sh --profile comet stop  
Stops the service for the comet profile.  
./modelersrv.sh stop  
Stops all active services
```

There is currently no supported method for starting SPSS Modeler Server profile services automatically on UNIX. The standard auto.sh script is available for configuring the system to start and stop the main SPSS Modeler Server service with the operating system, but this only applies to the default service -- not for any profile service.

Related information

- [Working in the enterprise](#)
- [Configuring single sign on](#)
- [Configuring groups](#)

Administration

IBM® SPSS® Modeler Server has a number of configurable options that control its behavior. You can set these options in two ways:

- Use the IBM SPSS Modeler Administration Console, which is available free of charge to current IBM SPSS Modeler customers. See the topic [IBM SPSS Modeler Server Administration](#) for more information.
- Use the options.cfg text file, located in the [server install path]/config directory. See the topic [Using the options.cfg file](#) for more information.

We recommend that you install IBM SPSS Deployment Manager and use its IBM SPSS Modeler Administration Console as your administration tool, rather than editing the options.cfg file. Editing the file requires access to the IBM SPSS Modeler Server file system, but IBM SPSS Modeler Administration Console allows you to authorize anyone with a user account to adjust these options. Also, IBM SPSS Modeler Administration Console provides additional information about the server processes, allowing you to monitor usage and performance. Unlike when editing the configuration file, most configuration options can be changed without restarting IBM SPSS Modeler Server.

More information about using IBM SPSS Modeler Administration Console and the options.cfg file is provided in the following topics.

Related information

- [IBM SPSS Modeler Server Administration](#)
- [Using the options.cfg file](#)

IBM SPSS Modeler Server Administration

The Modeler Administration Console in IBM® SPSS® Deployment Manager provides a console user interface to monitor and configure your SPSS Modeler Server installations, and is available free-of-charge to current SPSS Modeler Server customers. The application can be installed only on Windows computers; however, it can administer a server installed on any supported platform.

Many of the options available through the Modeler Administration Console can also be specified in the options.cfg file, which is located in the SPSS Modeler Server installation directory under /config. However, the Modeler Administration Console provides a shared graphical interface that allows you to connect, configure, and monitor multiple servers.

- [Starting Modeler Administration Console](#)
 - [Restarting the web service](#)
 - [Configuring Access with Modeler Administration Console](#)
 - [SPSS Modeler Server connections](#)
 - [SPSS Modeler Server Configuration](#)
 - [SPSS Modeler Server Monitoring](#)
 - [Using the options.cfg file](#)
 - [Closing unused database connections](#)
-

Starting Modeler Administration Console

From the Windows Start menu, choose [All] Programs, then IBM SPSS Collaboration and Deployment Services, then Deployment Manager.

When you first run the application, you see empty Server Administration and Properties panes (unless you already have Deployment Manager installed with an IBM® SPSS® Collaboration and Deployment Services server connection already set up). After you configure Modeler Administration Console, the Server Administrator pane on the left displays a node for each SPSS Modeler Server that you want to administer. The right-hand pane shows the configuration options for the selected server. You must first [set up a connection](#) for each server that you want to administer.

Restarting the web service

Whenever you make changes to an IBM® SPSS® Modeler Server in the Administration Console, you must restart the web service.

To restart the web service on Microsoft Windows:

1. On the computer where you installed IBM SPSS Modeler, select Services from Administrative Tools on the Control Panel.
2. Locate the server in the list and restart it.
3. Click OK to close the dialog box.

To restart the web service on UNIX:

On UNIX, you must restart the IBM SPSS Modeler Server by running the modelersrv.sh script in the IBM SPSS Modeler Server installation directory.

1. Change to the IBM SPSS Modeler Server installation directory. For example, at a UNIX command prompt, type:
cd /usr/<modelersrv>, where **modelersrv** is the IBM SPSS Modeler Server installation directory.
 2. To stop the server, at the command prompt, type
. ./modelersrv.sh stop
 3. To restart the server, at the command prompt, type
. ./modelersrv.sh start
-

Configuring Access with Modeler Administration Console

Administrator access to SPSS® Modeler Server through the Modeler Administration Console included with IBM® SPSS Deployment Manager is controlled with the **administrators** line in the options.cfg file, located in the SPSS Modeler Server installation directory under /config. This line is commented out by default, so you must edit this line to allow access to specific people, or use * to allow access to all users, as shown in the following examples:

```
administrators, "*"
administrators, "jsmith,mjones,achavez"
```

- The line must begin with **administrators**, and the entries must be contained in quotation marks. Entries are case sensitive.
- Separate multiple user IDs with commas.
- For Windows accounts, do not use domain names.
- Use the asterisk with care. It allows anyone with a valid user account for IBM SPSS Modeler Server (which, in most cases, is anyone on the network) to log in and change the configuration options.
- [Configuring Access with User Access Control](#)

Configuring Access with User Access Control

To use the Modeler Administration Console to make updates to a SPSS® Modeler Server configuration installed on a Windows machine that has User Access Control (UAC) enabled, you must have read, write, and execute permissions defined on the *config* directory and on the *options.cfg* file. These (NTFS) permissions must be defined at the specific user level and not at group level, this is due to the way that UAC and NTFS permissions interact.

The Modeler Administration Console is included in IBM® SPSS Deployment Manager.

Related information

- [Starting Modeler Administration Console](#)
- [SPSS Modeler Server connections](#)
- [SPSS Modeler Server Configuration](#)
- [Connections/Sessions](#)
- [Logging](#)
- [Data file access](#)
- [Performance/Optimization](#)
- [SQL](#)
- [SSL](#)
- [Coordinator of Processes configuration](#)
- [SPSS Modeler Server Monitoring](#)

SPSS Modeler Server connections

You must specify a connection to each SPSS® Modeler Server on your network that you want to administer. You must then log in to each server. Although the server connection is remembered across Modeler Administration Console sessions in IBM® SPSS Deployment Manager, the login credentials are not. You must log in every time you start IBM SPSS Deployment Manager.

To set up a server connection

1. Ensure that the IBM SPSS Modeler Server service is started.
2. From the File menu, choose New and then Administered Server Connection.
3. On the first page of the wizard, enter a name for your server connection. The name is for your own use and should be something descriptive; for example, *Production Server*. Ensure that Type is set to Administered **IBM SPSS Modeler Server**, then click Next.
4. On the second page, enter the hostname or IP address of the server. If you have changed the port from the default, enter the port number. Click Finish. The new server connection is shown in the Server Administrator pane.

To perform administration tasks, you must now log in.

To log in to the server

1. In the Server Administrator pane, double-click to select the server to which you want to log in.
2. In the Login dialog box, enter your credentials. (Use your user account for the server host.) Click OK.

If the login fails with the message Unable to obtain administrator rights on server, the most likely cause is that administrator access has not been configured correctly. See the topic [Configuring Access with Modeler Administration Console](#) for more information.

If the login fails with the message Failed to connect to server '<server>', make sure that the user ID and password are correct, then make sure that the IBM SPSS Modeler Server service is running. For example, under Windows, go to Control Panel > Administrative Tools > Services and check the entry for IBM SPSS Modeler Server. If the Status column does not show Started, select this line on the screen and click Start, then retry the login.

Once you log in to your IBM SPSS Modeler Server, two options are shown below the server name, [Configuration](#) and [Monitoring](#). Double-click one of these options.

- [Server Login Dialog Box](#)

Related information

- [Starting Modeler Administration Console](#)
- [Configuring Access with Modeler Administration Console](#)

- [SPSS Modeler Server Configuration](#)
 - [Connections/Sessions](#)
 - [Logging](#)
 - [Data file access](#)
 - [Performance/Optimization](#)
 - [SQL](#)
 - [SSL](#)
 - [Coordinator of Processes configuration](#)
 - [SPSS Modeler Server Monitoring](#)
-

Server Login Dialog Box

The Server Login dialog box allows you to log in to SPSS® Modeler Server in order to perform administration or monitoring tasks. Log in with the same network user account that you use to log in to IBM® SPSS Modeler Server from SPSS Modeler.

If you are unable to log in, contact your system administrator, who may need to grant you access.

SPSS Modeler Server Configuration

The Configuration pane shows configuration options for SPSS® Modeler Server. Use this pane to change the options as desired. Click Save on the toolbar to save the changes. Note that changing any option marked with an asterisk (*) requires a server restart to take effect.

The options are described in the following sections, with the corresponding line in options.cfg given in parentheses for each option. Options that are visible only in options.cfg are described at the end of this section.

Note: If a non-root user wants to change these options, write permission is required for the SPSS Modeler Server **config** directory.

- [Connections/Sessions](#)
- [Analytic Server connection](#)
- [Data file access](#)
- [Performance/Optimization](#)
- [SQL](#)
- [SSL](#)
- [Coordinator of Processes configuration](#)
- [Options visible in options.cfg](#)

Related information

- [Starting Modeler Administration Console](#)
 - [Configuring Access with Modeler Administration Console](#)
 - [SPSS Modeler Server connections](#)
 - [SPSS Modeler Server Monitoring](#)
-

Connections/Sessions

Modeler port number. (`port_number`) The port number for SPSS® Modeler Server to listen on. Change if another application already uses the default. End users must know the port number in order to use SPSS Modeler Server.

Embedded DataView service port number. (`data_view_port_number`) The port number for the DataView service embedded in SPSS Modeler Server to listen on. Change if another application already uses the default.

Maximum number of connections. (`max_sessions`) Maximum number of server sessions at one time. A value of -1 indicates no limit.

Analytic Server connection

Enable Analytic Server SSL (`as_ssl_enabled`). Specify Y to encrypt communications between Analytic Server, and SPSS® Modeler otherwise, N.

Host (`as_host`). The IP address of the Analytic Server.

Port Number (**as_port**). The Analytic Server port number.

Context Root (**as_context_root**). The context root of the Analytic Server.

Tenant (**as_tenant**). The tenant that the SPSS Modeler Server installation is a member of.

Realm (**as_realm**). The realm used for this Analytic Server.

Prompt for Password (**as_prompt_for_password**). Specify N if the SPSS Modeler Server is configured with the same authentication system for users and passwords as the system that is used on Analytic Server; for example, when you use Kerberos authentication, otherwise, Y.

Note: If you intend to use Kerberos SSO, you must set extra options in the options.cfg file. For more information, see [Options visible in options.cfg](#).

Note: To connect SSL enabled Analytic Server, extra steps are required as follows:

1. Use the following command to extract the certificate file trust.cer from the JKS file (i.e. trust.jks):

```
/bin/keytool -export -alias server-alias -storepass pass4jks -file /home/sslkeys/trust.cer -keystore /home/sslkeys/trust.jks
```

2. Import the trust.cer file into cacerts in the JRE used by your application server.
3. Import the trust.cer file into caserts in the JRE used by the SPSS Modeler Server.
4. Restart the SPSS Modeler Server and the IBM® SPSS Collaboration and Deployment Services Repository Server.

Data file access

Python executable path for Extension nodes and Custom Dialog Builder. (**eas_pyspark_python_path**) Full path to the Python executable file, including the name of the file. [**program_files_restricted**] may need to be set to No depending on your Python installation location.

Restrict access to only data file path. (**data_files_restricted**) When set to yes, this option restricts data files to the standard data directory and those listed in the Data File Path below.

Data file path. (**data_file_path**) A list of additional directories to which clients are allowed to read and write data files. This option is ignored unless the Restrict Access to Only Data File Path option is turned on. Note that you should use forward slashes in all path names. On Windows, specify multiple directories using semicolons (for example, [server install path]/data;c:/data;c:/temp). On Linux and UNIX, use colons (:) instead of semicolons. The data file path must include any path(s) specified by the **temp_directory** parameter described below.

Restrict access to only program files path. (**program_files_restricted**) When set to yes, this option restricts program file access to the standard bin directory and those listed in the Program files path below. As of release 17, the only program file to which access is restricted is the Python executable (see the Python executable path below).

Program files path. (**program_file_path**) A list of additional directories from which clients are allowed to execute programs. This option is ignored unless the Restrict Access to Only Program Files Path option is turned on. Note that you should use forward slashes in all pathnames. Specify multiple directories using semicolons.

Maximum file size. (**max_file_size**) Maximum size (in bytes) of temporary and exported data files created during stream execution (does not apply to SAS and SPSS® Statistics data files). A value of -1 indicates no limit.

Temporary directory. (**temp_directory**) The directory used to store temporary data files (cache files). Ideally, this directory should be on a separate high-speed drive or controller because speed of access to this directory may have a significant impact on performance. You may specify multiple temporary directories, separating each with a comma (for example: **temp_directory**, "D:/Modeler_temp, C:/Program Files/IBM/SPSS/ModelerServer/<version>/Tmp"). These should be located on different disks; the first directory is used most often, and additional directories are used to store temporary work files when certain data preparation operations (such as sort) use parallelism during execution. Allowing each execution thread to use separate disks for temporary storage can improve performance. Use forward slashes in all path specifications.

Note:

- Temporary files are generated in this directory during startup of SPSS Modeler Server. Ensure that you have the necessary access rights to this directory (for example, if the temporary directory is a shared network folder), otherwise SPSS Modeler Server startup will fail.
- The **temp_directory** setting does not apply when running Evaluation streams via IBM® SPSS Collaboration and Deployment Services jobs. When you run such a job, a temporary file is created. By default, the file is saved to the IBM SPSS Modeler Server installation directory. You can change the default data folder that the temp files are saved to when you create the IBM SPSS Modeler Server connection in IBM SPSS Modeler.

Python executable path for bulk loading. (**python_exe_path**) Full path to the Python executable including the executable name. If access to program files is restricted, then you must add the directory containing the Python executable to the program files path (see Restrict access to only program files path above).

Path to OPL library of full version of CPLEX. (**cplex_opl_lib_path**) Path to the OPL library for the full version of CPLEX.

Performance/Optimization

Stream rewriting. (`stream_rewriting_enabled`) Allows the server to optimize streams by rewriting them. For example, the server might push data reduction operations closer to the source node to minimize the size of the dataset as early as possible. Disabling this option is normally recommended only if the optimization causes an error or other unexpected results. This setting overrides the corresponding client optimization setting. If this setting is disabled in the server, then the client cannot enable it. But if it is enabled in the server, the client can choose to disable it.

Parallelism. (`max_parallelism`) Describes the number of parallel worker threads that SPSS® Modeler is allowed to use when running a stream. Setting this to 0 or any negative number causes IBM® SPSS Modeler to match the number of threads to the number of available processors on the computer; the default value for this option is -1. To turn off parallel processing (for machines with multiple processors), set this option to 1. To allow limited parallel processing, set it to a number smaller than the number of processors on your machine. Note that a hyperthreaded or dual-core processor is treated as two processors.

Buffer size (bytes). (`io_buffer_size`) Data files transferred from the server to the client are passed through a buffer of this number of bytes.

Cache compression. (`cache_compression`) An integer value in the range 0 to 9 that controls the compression of cache and other files in the server's temporary directory. Compression reduces the amount of disk space used, which can be important when space is limited. Compression increases processor time, but this is almost always made up by the reduction in disk access time. Note that only certain caches, those accessed sequentially, can be compressed. This option does not apply to random-access caches, such as those used by the network training algorithms. A value of 0 disables compression entirely. Values from 1 upward provide increasing degrees of compression but with a corresponding cost in access time. The default value is 1; higher values may be needed where disk space is at a premium.

Memory usage multiplier. (`memory_usage`) Controls the proportion of physical memory allocated for sorting and other in-memory caches. The default is 100, which corresponds to approximately 10% of physical memory. Increase this value to improve sort performance where free memory is available, but be careful of increasing it so high as to cause excessive paging.

Modeling memory limit percentage. (`modelling_memory_limit_percentage`) Controls the proportion of physical memory allocated for training Kohonen and *k*-means models. The default is 25%. Increase this value to improve training performance where free memory is available, but be careful of increasing it so high as to cause excessive paging when data spills onto the disk.

Allow modeling memory override. (`allow_modelling_memory_override`) Enables or disables the Optimize for Speed option in certain modeling nodes. The default is enabled. This option allows the modeling algorithm to claim all available memory, bypassing the percentage limit option. You may want to disable this if you need to share memory resources on the server machine.

Maximum and minimum server port. (`max_server_port` and `min_server_port`) Specifies the range of port numbers that can be used for the additional socket connections between client and server that are required for interactive models and stream execution. These require the server to listen on another port; not restricting the range could cause problems for users on systems with firewalls. Default value for both is -1, meaning "no restriction." Thus, for example, to set the server to listen on port 8000 or above, you would set `min_server_port` to 8000 and `max_server_port` to -1.

Note that you must open additional ports over the main server port to open or execute a stream, and correspondingly more ports if you want to open or execute concurrent streams. This is required in order to capture feedback from the stream execution.

By default, IBM SPSS Modeler will use any open port that is available; if it does not find one (for example, if they are all closed by a firewall), an error is displayed when you execute the stream. To configure the range of ports, IBM SPSS Modeler will need two open ports (in addition to the main server port) available per concurrent stream, plus 3 additional ports for each ODBC connection from within any connected client (2 ports for the ODBC connection for the duration of that ODBC connection, and an additional temporary port for authentication).

Note: An ODBC connection is an entry in the database connections list, and can be shared between multiple database nodes specified with the same database connection.

Note: It is possible that the authentication ports can be shared if the connections are made at different times.

Note: Best practice dictates that the same ports should be used to communicate with both IBM SPSS Collaboration and Deployment Services and SPSS Modeler Client. These can be set as `max_server_port` and `min_server_port`.

Note: If you change these parameters, you need to restart SPSS Modeler Server for the change to take effect.

Array fetch optimization. (`sql_row_array_size`) Controls the way that SPSS Modeler Server fetches data from the ODBC datasource. The default value is 1, which fetches a single row at a time. Increasing this value causes the server to read the information in larger chunks, fetching the specified number of rows into an array. With some operating system/database combinations, this can result in improvements to the performance of `SELECT` statements.

Related information

- [IBM SPSS Modeler Server Administration](#)
- [Starting Modeler Administration Console](#)
- [Configuring Access with Modeler Administration Console](#)
- [SPSS Modeler Server connections](#)
- [Connections/Sessions](#)
- [Logging](#)
- [Data file access](#)

- [SQL](#)
 - [SSL](#)
 - [Coordinator of Processes configuration](#)
 - [SPSS Modeler Server Monitoring](#)
-

SQL

Maximum SQL string length. (**max_sql_string_length**) For a string imported from the database with SQL, the maximum number of characters that are guaranteed to be passed successfully. Depending on the operating system, string values longer than this may be truncated on the right without warning. The valid range is between 1 and 65,535 characters. This property also applies to the Database export node.

Note: The default value for this parameter is 2048. If the text you are analyzing is longer than 2048 characters (for example, this may occur if using the SPSS® Modeler Text Analytics Web Feed node) we recommend increasing this value if working in native mode otherwise your results may be truncated. If you are using a database and user-defined functions (UDF), this restriction does not occur; this can account for differences in results between native and UDF modes.

Automatic SQL generation. (**sql_generation_enabled**) Allows automatic SQL generation for streams, which may substantially improve performance. The default is enabled. Disabling this option is recommended only if the database is not able to support queries submitted by SPSS Modeler Server. Note that this setting overrides the corresponding client optimization setting; also note that for purposes of scoring, SQL generation must be enabled separately for each modeling node regardless of this setting. If this setting is disabled in the server, then the client cannot enable it. But if it is enabled in the server, the client can choose to disable it.

Default SQL string length. (**default_sql_string_length**). Specifies the default width of string columns that will be created within database cache tables. String fields in database cache tables will be created with a default width of 255 if there is no upstream type information. If you have wider values than this in your data, either instantiate an upstream Type node with those values, or set this parameter to a value that is large enough to accommodate those string values.

Enable Database UDF. (**db_udf_enabled**). If set to **y** (default), causes the SQL generation option to generate user-defined function (UDF) SQL instead of pure SPSS Modeler SQL. UDF SQL usually outperforms pure SQL.

SSL

Enable SSL. (**ssl_enabled**) Enables SSL encryption for connections between SPSS® Modeler and SPSS Modeler Server.

SSL keystore file. (**ssl_keystore**) The SSL key database file to be loaded when the server starts (either a full path or a relative path to the SPSS Modeler installation directory).

SSL keystore stash file. (**ssl_keystore_stash_file**) The name of the key database password stash file to be loaded when the server starts up (either a full path or a relative path to the SPSS Modeler installation directory). On Linux, if you want to leave this setting blank and be prompted for the password when starting the SPSS Modeler Server, see the following instructions:

- On Linux/UNIX:
 1. Make sure the **ssl_keystore_stash_file** setting in options.cfg does not have a value.
 2. Locate the following line in the modelersrv.sh file:

```
if "$INSTALLEDPATH/$SCLEMNAME" -server $ARGS; then
```
 3. Add the **-request_ssl_password** switch as follows:

```
if "$INSTALLEDPATH/$SCLEMNAME" -request_ssl_password -server $ARGS;
then
```
 4. Restart the SPSS Modeler Server. You will be prompted for a password. Enter the correct password, click OK, and the server will start.

Keystore certificate label. (**ssl_keystore_label**) Label for the specified certificate.

Note: To use the Administration Console with a server setup for SSL, you must import any certificates required by SPSS Modeler Server into the Deployment Manager trust store (under/jre/lib/security).

Note: If you change these parameters, you need to restart SPSS Modeler Server for the change to take effect.

Coordinator of Processes configuration

Host. (**cop_host**) The hostname or IP address of the Coordinator of Processes service. The default "spsscop" is a vanity name which administrators can choose to add as an alias for the IBM® SPSS® Collaboration and Deployment Services host in DNS.

Port number. (**`cop_port_number`**) The port number of the Coordinator of Processes service. The default, 8080, is the IBM SPSS Collaboration and Deployment Services default.

Context root. (**`cop_context_root`**) The URL of the Coordinator of Processes service.

Login name. (**`cop_user_name`**) The user name for authentication to the Coordinator of Processes service. This is an IBM SPSS Collaboration and Deployment Services login name so may include a security-provider prefix (for example: ad/jsmith).

Password. (**`cop_password`**) The password for authentication to the Coordinator of Processes service.

Note: If you update the options.cfg file manually instead of using the Modeler Administration Console in IBM SPSS Deployment Manager, you must manually encode the **`cop_password`** value that you specify in the file. Plain text passwords are invalid and cause registration with the Coordinator of Processes to fail.

Follow these steps to manually encode the password:

1. Open a Command Prompt, navigate to the SPSS Modeler .**/bin** directory, and run the command **pwutil.bat/sh**.
2. When requested, type the user name (the **`cop_user_name`** you are specifying in options.cfg) and press Enter.
3. When requested, type the password for that user.

The encoded password is displayed between double quotes on the command line as part of the returned string. For example:

```
C:\Program Files\IBM\SPSS\Modeler\18\bin>pwutil
User name: copuser
Password: Pass1234
copuser, "0Tqb4n.ob0wrs"
```

4. Copy the encoded password, without the double quotes, and paste it between the double quotes that already exist for the **`cop_password`** value in the options.cfg file.

Enabled. (**`cop_enabled`**) Determines whether the server should attempt to register with the Coordinator of Processes. The default is *not* to register because the administrator should choose which services are advertised through the Coordinator of Processes.

SSL Enabled. (**`cop_ssl_enabled`**) Determines whether SSL is used to connect to the Coordinator of Processes server. If this option is used, you must import the SSL certificate file into the SPSS Modeler Server JRE. To do this, you must obtain the SSL certificate file and its alias name and password. Then run the following command on the SPSS Modeler Server:

```
$JAVA_HOME/bin/keytool -import -trustcacerts -alias $ALIAS_NAME -file
$CERTIFICATE_FILE_PATH -keystore $ModelerServer_Install_Path/jre/lib/security/cacerts
```

Server name. (**`cop_service_name`**) The name of this SPSS Modeler Server instance; the default is the host name.

Description. (**`cop_service_description`**) A description of this instance.

Update interval (min). (**`cop_update_interval`**) The number of minutes between keep-alive messages; the default is 2.

Weight. (**`cop_service_weight`**) The weight of this instance, specified as an integer between 1 and 10. A higher weight attracts more connections. The default is 1.

Service host. (**`cop_service_host`**) The fully-qualified host name of the IBM SPSS Modeler Server host. The default of the host name is derived automatically; the administrator can override the default for multi-homed hosts.

Default data path. (**`cop_service_default_data_path`**) The default data path for a Coordinator of Processes registered IBM SPSS Modeler Server installation.

Related information

- [IBM SPSS Modeler Server Administration](#)
- [Starting Modeler Administration Console](#)
- [Configuring Access with Modeler Administration Console](#)
- [SPSS Modeler Server connections](#)
- [Connections/Sessions](#)
- [Logging](#)
- [Data file access](#)
- [Performance/Optimization](#)
- [SQL](#)
- [SSL](#)
- [SPSS Modeler Server Monitoring](#)

Options visible in options.cfg

Most configuration options can be changed using the IBM® SPSS® Modeler Administration Console included with IBM SPSS Deployment Manager. But there are some exceptions, such as those described in this section. The options in this section must be changed by editing the options.cfg file. See [IBM SPSS Modeler Server Administration](#) and [Using the options.cfg file](#) for more information. Note that there may be additional settings in options.cfg that are not listed here.

Note: This information only applies to a remote server (IBM SPSS Modeler Server, for example).

administrators. Specify the user names of those users to whom you want to grant administrator access. See the topic [Configuring Access with Modeler Administration Console](#) for more information.

allow_config_custom_overrides. Do not modify unless instructed to do so by a technical-support representative.

data_view_port_number. You can right-click a data node and select View Data to examine and refine your data in interesting ways with advanced data visualizations. This feature uses the port number 28900 by default. Modify the value for this **data_view_port_number** configuration option if you need to use a different port number. We recommend using the default, if possible.

fips_encryption. Enables FIPS compliant encryption. The default is **N**.

group_configuration. When enabled, IBM SPSS Modeler Server checks the groups.cfg file which controls who can log on to the server.

max_transfer_size. For internal system use only. **Do not modify.**

shell. (UNIX servers only) Overrides the default setting for the UNIX shell, for example **shell**, "/usr/bin/ksh". By default, IBM SPSS Modeler uses the shell defined in the user profile of the user who is connecting to IBM SPSS Modeler Server.

start_process_as_login_user. Set this to **y** if you are running SPSS Modeler Server with a private password database, starting the server service from a non-root account.

use_bigint_for_count. When the number of the records to be counted is larger than a normal integer ($2^{31}-1$) can hold, set this option to **y**. When this option is set to **y**, and a stream is connected to Db2; SQL Server; or a Teradata, Oracle, or Netezza database, a function is used where a record count is needed (for example, the Record_Count field generated by the Aggregate node).

When this option is enabled, and if working with either Db2 or SQL Server, SPSS Modeler uses COUNT_BIG() for record counting. If working with Teradata, Oracle, or Netezza, SPSS Modeler will use COUNT(). For all other databases, there is no SQL pushback for the function. The difference is that when **use_bigint_for_count** is enabled, all record counts are saved as BIG INT (or LONG) type (a 64-bit signed integer, $2^{63}-1$ to the maximum), as compared to normal integer (a 32-bit signed integer, $2^{31}-1$ to the maximum) when the options is disabled.

cop_ssl_enabled. Set this option to **y** if you are using SSL to connect to the Coordinator of Processes Service. If this option is used, you must import the SSL certificate file into the SPSS Modeler Server JRE. To do this, you must obtain the SSL certificate file and its alias name and password. Then run the following command on the SPSS Modeler Server:

```
$JAVA_HOME/bin/keytool -import -trustcacerts -alias $ALIAS_NAME -file  
$CERTIFICATE_FILE_PATH -keystore $ModelerServer_Install_Path/jre/lib/security/cacerts
```

cop_service_default_data_path. You can use this option to set the default data path for a Coordinator of Processes registered IBM SPSS Modeler Server installation.

Users can create their own Analytic Server connections in SPSS Modeler via Tools > Analytic Server Connections. Administrators can also define a default Analytic Server connection using the following properties:

as_ssl_enabled. **Y** or **N**.

as_host. Specify the Analytic Server host name or IP address.

as_port. Specify the Analytic Server port number.

as_context_root. Specify the Analytic Server context root.

as_tenant. Specify the name of the tenant that the IBM SPSS Modeler Server is a member of

as_prompt_for_password. **Y** or **N**.

By default, Analytic Server authentication using the Kerberos method is not enabled. To enable Kerberos authentication, use the three following properties:

as_kerberos_auth_mode. To enable Kerberos authentication, set this option to **y**.

as_kerberos_krb5_conf. Specify the path to the Kerberos configuration file that Analytic Server should use; for example, c:\windows\krb5.conf.

as_kerberos_krb5_spn. Specify the Analytic Server Kerberos SPN; for example, HTTP/ashost.mydomain.com@MYDOMAIN.COM.

SPSS Modeler Server Monitoring

The monitoring pane of Modeler Administration Console in IBM® SPSS® Deployment Manager shows a snapshot of all processes running on the SPSS Modeler Server computer, similar to the Windows Task Manager. To activate the monitoring pane, double-click the Monitoring node beneath the desired server in the Server Administrator pane. This populates the pane with a current snapshot of data from the server. The data refreshes at the rate shown (one minute by default). To refresh the data manually, click the Refresh button. To show only SPSS Modeler Server processes in this list, click the Filter out non-**SPSS Modeler** processes button.

Related information

- [Starting Modeler Administration Console](#)
 - [Configuring Access with Modeler Administration Console](#)
 - [SPSS Modeler Server connections](#)
 - [SPSS Modeler Server Configuration](#)
 - [Connections/Sessions](#)
 - [Logging](#)
 - [Data file access](#)
 - [Performance/Optimization](#)
 - [SQL](#)
 - [SSL](#)
 - [Coordinator of Processes configuration](#)
-

Using the options.cfg file

The options.cfg file is located in the [server install path]/config directory. Each setting is represented by a comma-separated name-value pair, where the **name** is the name of the option and the **value** is the value for the option. Pound (hash) signs (#) indicate comments.

Note: Most configuration options can be changed using IBM® SPSS® Modeler Administration Console in IBM SPSS Deployment Manager, rather than this configuration file, but there are a few exceptions. See the topic [Options visible in options.cfg](#) for more information.

By using IBM SPSS Modeler Administration Console, you can avoid server restarts for all options except the server port. See the topic [IBM SPSS Modeler Server Administration](#) for more information.

Note: This information only applies to a remote server (IBM SPSS Modeler Server, for example).

Configuration options that can be added to the default file

By default, in-database caching is enabled with IBM SPSS Modeler Server. You can disable this feature by adding the following line to the options.cfg file:

```
enable_database_caching, N
```

Doing so causes temporary files to be created on the server and not in the database.

To view or change the IBM SPSS Modeler Server configuration options:

1. Open the options.cfg file with a text editor.
 2. Locate the options of interest. For a full list of options, see [SPSS Modeler Server Configuration](#).
 3. Edit the values, as appropriate. Note that all pathname values must use a forward slash (/) rather than a backslash as the pathname separator.
 4. Save the file.
 5. Stop and restart IBM SPSS Modeler Server so that the changes will take effect.
-

Closing unused database connections

By default, IBM® SPSS® Modeler caches at least one connection to a database once that connection has been accessed. The database session is held open even when streams requiring database access are not being executed.

Caching database connections can improve execution times by removing the need for IBM SPSS Modeler to reconnect to the database each time a stream is executed. However, in some environments, it is important for applications to release database resources as quickly as possible. If too many IBM SPSS Modeler sessions maintain connections to the database that are no longer used, database resources may become exhausted.

You can avoid this possibility by turning off the IBM SPSS Modeler option **cache_connection** in a custom database configuration file. Doing so can also make IBM SPSS Modeler more resilient to faults in the database connection (such as timeouts) that can occur when connections are used over a long period of time by an IBM SPSS Modeler session.

To cause unused database connections to be closed:

1. Locate the `[server install path]/config` directory.
2. Add the following file (or open it, if it already exists):
`odbc-custom-properties.cfg`
3. Add the following line to the file:
`cache_connection, N`
4. Save and close the file.
5. Restart IBM SPSS Modeler Server.

Note:

In-database caches are saved in database as either a regular table or a temporary table, depending on each database's implementation. For example, temporary tables are used for Db2, Oracle, Amazon Redshift, Sybase, and Teradata. For these databases, setting `cache_connection` to `N` does not work as expected because the temporary table is only valid within a session (it will be cleaned automatically by the database when the database connection is closed).

So when running an SPSS Modeler stream against one of these databases with `cache_connection` set to `N`, an error such as Failed to create table for in-database caching. Using file cache instead. may result. This indicates that SPSS Modeler failed to created the in-database cache. Also, in some cases for an SPSS Modeler-generated SQL query, a temporary table is used but the table is empty.

To work around this issue, you can choose to use a regular database table for in-database caches. To do this, create a custom database property configuration file that contains the following line:

```
table_create_temp_sql, 'CREATE TABLE <table-name> <(table-columns)>'
```

This forces a regular database table to be used for the in-database cache, and the table will be dropped when all connections to the database are closed or when the working stream is closed.

Using SSL to secure data transfer

Secure Sockets Layer (SSL) is a protocol for encrypting data transferred between two computers. SSL ensures that communication between the computers is secure. SSL can encrypt the authentication of a username/password and the contents of an exchange between a server and client.

- [How SSL works](#)
SSL relies on the server's public and private keys, in addition to a public key certificate that binds the server's identity to its public key.
- [Securing client/server and server-server communications with SSL](#)
Securing communications with SSL requires a number of configuration steps on the client and server systems.
- [Cognos SSL connection](#)
To enable SSL access between SPSS Modeler and Cognos Analytics, you must create and import the relevant certificate.
- [Cognos TM1 SSL connection](#)
To enable SSL access between SPSS Modeler and Cognos TM1, you must create and import the relevant certificate.
- [Configuring the URL prefix](#)
You must configure SPSS Collaboration and Deployment Services Repository URL prefix to enable SSL access to the repository.
- [Securing LDAP with SSL](#)
The following example illustrates how to enable SSL for LDAP with Microsoft Active Directory as a security provider.

How SSL works

SSL relies on the server's public and private keys, in addition to a public key certificate that binds the server's identity to its public key.

1. When a client connects to a server, the client authenticates the server with the public key certificate.
2. The client then generates a random number, encrypts the number with the server's public key, and sends the encrypted message back to the server.
3. The server decrypts the random number with its private key.
4. From the random number, both the server and client create the session keys used for encrypting and decrypting subsequent information.

The public key certificate is typically signed by a certificate authority. Certificate authorities, such as VeriSign and Thawte, are organizations that issue, authenticate, and manage security credentials contained in the public key certificates. Essentially, the certificate authority confirms the identity of the server. The certificate authority usually charges a monetary fee for a certificate, but self-signed certificates can also be generated.

Securing client/server and server-server communications with SSL

The main steps in securing client/server and server-server communications with SSL are:

1. Obtain and install the SSL certificate and keys.
2. Enable and configure SSL in the server administration application (IBM® SPSS® Deployment Manager).
3. If using encryption certificates with a strength greater than 2048 bits, install unlimited strength encryption on the client computers.
4. If using a self-signed certificate, copy the certificate on the client computer.
5. Instruct users to enable SSL when connecting to the server.

Notes:

- Occasionally a server product acts as a client. An example is IBM SPSS Statistics Server connecting to the IBM SPSS Collaboration and Deployment Services Repository. In this case, IBM SPSS Statistics Server is the *client*.
- On Linux/UNIX systems where both a non-root and SSL configuration are enabled, SSL security will be reduced. Because all user sessions run under the same credential as each other and as the Modeler Server daemon, the SSL certificate data that should be kept secret will instead be exposed to all users. This allows users to easily bypass the normal protections SSL provides to all other users. See [Introduction](#).
- **[Obtaining and installing SSL certificate and keys](#)**
For IBM SPSS Modeler; obtain or generate the certificate and key files and install them on the server.
- **[Enable and configure SSL in IBM SPSS Deployment Manager](#)**
Perform the steps to enable SSL for server administration in SPSS Deployment Manager.
- **[Installing unlimited strength encryption](#)**
The Java Runtime Environment shipped with the product has US export-strength encryption enabled. For enhanced security of your data, upgrading to unlimited-strength encryption is recommended. This procedure must be repeated for both client and server installations.
- **[Instructing users to enable SSL](#)**
When you connect to the server through a client product, enable SSL in the server login dialog box.

Obtaining and installing SSL certificate and keys

The first steps you must follow to configure SSL support are:

1. Obtain an SSL certificate and key file. There are three ways you can do this:
 - Purchase them from a public certificate authority (such as VeriSign, Thawte, or Entrust). The public certificate authority (CA) signs the certificate to verify the server that uses it.
 - Generate the key and certificate files with a third-party certificate authority. If this approach is taken then the third-party CA's root certificate must be imported into the client and server keystore files. See the topic [Importing a third-party root CA certificate](#) for more information.
 - Generate the key and certificate files with an internal self-signed certificate authority. The steps to do this are:
 - Prepare a key database. See the topic [Creating an SSL key database](#) for more information.
 - Create the self-signed certificate. See the topic [Creating a self-signed SSL certificate](#) for more information.
2. Copy the .kdb and .sth files created in step 1 into a directory to which the IBM SPSS Modeler Server has access and specify the path to that directory in the options.cfg file..
Note: Use forward slashes as separators in the directory path.
3. Set the following parameters in the options.cfg file:
 - ssl_enabled, Y
 - ssl_keystore, "<filename>.kdb" where <filename> is the name of your key database.
 - ssl_keystore_stash_file, "<filename>.sth" where <filename> is the name of the key database password stash file.
 - ssl_keystore_label, <label> where <label> is the label of your certificate.
4. For self-signed or third-party certificates install the certificate on client systems. For purchased public CA certificates, this step is not required. Ensure that access permissions deny casual browsing of the directory that contains the certificate. See the topic [Installing a self-signed SSL certificate](#) for more information.
 - [Configuring the environment to run GSKit](#)
 - [Creating an SSL key database](#)
 - [Creating a self-signed SSL certificate](#)
 - [Installing a self-signed SSL certificate](#)
 - [Importing a third-party root CA certificate](#)

Configuring the environment to run GSKit

The GSKCapiCmd is a non-Java-based command-line tool, and Java™ does not need to be installed on your system to use this tool; it is located in the <Modeler installation directory>/bin folder. The process to configure your environment to run IBM Global Security Kit (GSKit) varies depending on the platform in use.

To configure for Linux/Unix, add the shared libraries directory <Modeler installation directory>/lib to your environment:

```
$export <Shared library path environment variable>=<modeler_server_install_path>/bin  
$export PATH=$PATH:<modeler_server_install_path>/bin
```

The shared library path variable name depends on your platform:

- HP-UX uses the variable name: **SHLIB_PATH**
- Linux uses the variable name: **LD_LIBRARY_PATH**

For example, to set the environment on Linux, use:

```
$export LD_LIBRARY_PATH=/path/to/gskit/bin  
$export PATH=$PATH:/path/to/gskit/bin
```

Account access to files

Ensure that you grant the correct permissions for the accounts that will access the SSL files:

1. For all accounts that are used by SPSS® Modeler for connection, grant read access to the SSL files.
Note: This also applies to the *Log on as* user that is defined in the SPSS Modeler Server service. On UNIX or Linux, it applies to the user you are starting the server as.
2. For Windows, it is not enough that the accounts are in the Administrators group and that permission is given to that Administrators group when User Access Control (UAC) is enabled. In addition you must take one of the following actions:
 - Give the accounts permission separately.
 - Create a new group, add the accounts into the new group, and give the group permission to access the SSL files.
 - Disable UAC.

Creating an SSL key database

Use the GSKCapiCmd tool to create your key database. Before using the tool, you must configure your environment; see the topic [Configuring the environment to run GSKit](#) for more information

To create the key database, run GSKit and enter the following command:

```
gsk<ver>capi cmd[_64] -keydb -create -populate -db <filename>.kdb -pw <password> -stash
```

where **<ver>** is the GSKit version number, **<filename>** is the name you want to use for the key database file, and **<password>** is the password for the key database.

The **-stash** option creates a stash file at the same path as the key database, with a file extension of .sth. GSKit uses the stash file to obtain the password to the key database so that it doesn't have to be entered on the command line each time.

Note: You should use strong file system protection on the .sth file.

Creating a self-signed SSL certificate

To generate a self-signed certificate and store it in the key database, use the following command:

```
gsk<ver>capi cmd[_64] -cert -create -db <filename>.kdb -stashed -dn "CN=myserver,OU=mynetwork,O=mycompany,C=mycountry" -label <label> -expire <Number of days certificate is valid> -default_cert yes
```

where **<ver>** is the GSKit version number, **<filename>** is the name of the key database file, **<Number of days certificate is valid>** is the physical number of days that the certificate is valid, and **<label>** is a descriptive label to help you identify the file (for example, you could use a label such as: **myselfsigned**).

Installing a self-signed SSL certificate

For the client machines that connect to your server using SSL, you must distribute the public part of the certificate to the clients so that it can be stored in their key databases. To do this, perform the following steps:

1. Extract the public part to a file using the following command:

```
gsk<ver>capi cmd[_64] -cert -extract -db <filename>.kdb -stashed -label <label> -format ascii -target mycert.arm
```

2. Distribute mycert.arm to the clients. It should be copied to their jre/bin directory.
3. Add the new certificate to the clients' key database using the following command:

```
keytool -import -alias <label> -keystore ..\lib\security\cacerts -file mycert.arm
```

If prompted for a password, use: `changeit`. The keytool is located in the <Modeler installation directory>\jre\bin directory (or in the <Modeler installation directory>/SPSSModeler.app/Contents/PlugIns/jre/Contents/Home/bin directory on Mac).

Importing a third-party root CA certificate

Instead of purchasing a certificate from a well known certificate authority (CA) or creating a self-signed certificate, you can use a third-party certificate authority to sign your server certificates. The client and server must have access to the third-party CA's root certificate to verify the server certificates that are signed by the third-party CA. To do this:

1. Obtain the third-party CA root certificate. The process for this varies depending on the third-party CA's procedures. Third-party CAs often make their root certificates available for download.
2. Add the certificate to the servers' key database using the following command:

```
gsk<ver>capicmd[ _64} -cert -add -db <filename>.kdb -stashed -label <label> -file <ca_certificate>.crt  
-format binary -trust enable
```

3. Add the certificate to the clients' key database using the following command:

On Windows:

```
C:> cd <Modeler Client installation path>\jre\bin  
C:> keytool -import -keystore ..\lib\security\cacerts -file <ca_certificate>.crt -alias <label>
```

On Mac:

```
C:> cd <Modeler Client installation path>/SPSSModeler.app/Contents/PlugIns/jre/Contents/Home/bin  
C:> keytool -import -keystore ..\lib\security\cacerts -file <ca_certificate>.crt -alias <label>
```

If prompted for a password, use: `changeit`. The keytool is located in the <Modeler installation directory>\jre\bin directory (or in the <Modeler installation directory>/SPSSModeler.app/Contents/PlugIns/jre/Contents/Home/bin directory on Mac).

4. Validate the server's key database with the root CA certificate using the following command:

```
gsk<ver>capicmd[ _64} -cert -validate -db <filename>.kdb -stashed -label <label>
```

A successful validation is indicated by the returned message: OK.

Note: The commands explained above use a third-party CA root certificate that is in a binary format. If the certificate is in an ASCII format, use the `-format ascii` option.

The `-db` parameter specifies the name of the key database into which you import the third-party CA root certificate.

The `-label` parameter specifies the label to use for the third-party CA root certificate inside the key database file. The label you use here can be anything because it does not have any relation to the labels used in the IBM® SPSS® Modeler options.cfg file.

The `-file` parameter specifies the file that contains the third-party CA root certificate

Enable and configure SSL in IBM SPSS Deployment Manager

1. If installing a self-signed SSL certificate, copy the cacerts file that you created to the <Deployment Manager installation directory>\jre\lib\security directory. See the topic [Installing a self-signed SSL certificate](#) for more information.
2. Start the server administration application (IBM® SPSS® Deployment Manager) and connect to the server.
3. On the configuration page, set Secure Sockets Layer to `Yes`.
4. In SSL Public Key File, specify the full path to the public key file.
5. In SSL Private Key File, specify the full path to the private key file.
Note: If the public and private keys are stored in one file, specify the same file in SSL Public Key File and SSL Private Key File.
6. From the menus choose:
`File > Save`
7. Restart the server service or daemon. When you restart, you will be prompted for the SSL password. On Windows, you can select Remember this password to store the password securely. This option eliminates the need to enter the password every time the server is started.

Installing unlimited strength encryption

The Java Runtime Environment shipped with the product has US export-strength encryption enabled. For enhanced security of your data, upgrading to unlimited-strength encryption is recommended. This procedure must be repeated for both client and server installations.

To install unlimited strength encryption

1. Download the [Unrestricted SDK JCE policy files](#) from IBM.com (select the files applicable to Java 8).
Note: You will need to login with your IBMid credentials in order to download the files.
 2. Extract the unlimited jurisdiction policy files that are packaged in the compressed file. The compressed file contains a *US_export_policy.jar* file and a *local_policy.jar* file.
 3. Back up the existing copies of *US_export_policy.jar* and *local_policy.jar* from the directory *jre/lib/security*.
 4. Replace the existing copies of *US_export_policy.jar* and *local_policy.jar* files with the two files that you downloaded and extracted.
 5. Restart IBM® SPSS® Modeler Client or Server as appropriate.
-

Instructing users to enable SSL

When users connect to the server through a client product, they need to enable SSL in the dialog box for connecting to the server.

Cognos SSL connection

To connect to a Cognos Analytics server with HTTPS and an SSL secured port, you must first change some of the Cognos internal and external dispatcher settings. For details on how to make the required changes, see the Cognos Server Configuration and Administration guide.

After you change the dispatcher settings, import the SSL certification that you created in Cognos into the SPSS® Modeler JRE by following these steps:

1. In Cognos configuration, define a password for the IBM Cognos key store:
 - a. In the Explorer window, click *Cryptography* > *Cognos*.
 - b. In the Properties window, under *Encryption Key Settings*, set the *Encryption key store password*.
 - c. From the *File* menu, select *Save*.
 - d. From the *Actions* menu, select *Restart*.
2. From the command line, go to the *c10_location\bin* directory.
3. Set the *JAVA_HOME* environment variable to the Java™ Runtime Environment location used by the application server that is running Cognos. For example:

```
set JAVA_HOME=c11_location\bin\jre\<version>
```

4. From the command line, run the certificate tool. For example:

```
ThirdPartyCertificateTool.bat -E -T -r ca.cer -k ..\configuration\encryptkeypair\jEncKeystore  
-p <password>
```

5. Copy the *ca.cer* file to the SPSS Modeler Server location.
6. Open a command line and switch to the *<ModelerInstallationLocation>\jre\bin* folder.
7. Run the command to import the certificate. For example:

```
.\keytool -import -alias ca -file <Directory where ca.cer is located>\ca.cer  
-keystore "<ModelerInstallationLocation>\jre\lib\security\cacerts"
```

You can then use HTTPS and the SSL secured dispatcher to connect to Cognos. For example:

```
https://9.119.83.37:9343/p2pd/servlet/dispatch
```

Cognos TM1 SSL connection

To connect to Cognos TM1 with HTTPS and an SSL secured port, follow these steps:

1. Enable SSL on IBM Cognos TM1 Application Server. See the TM1 documentation at <https://www.ibm.com/docs/en/cognos-tm1/10.2.2?topic=configuration-usingssl-data-transmission-security>.
2. Download the *tm1server.pem* certification file. For example, if using Firefox:
 - a. Open a browser window and enter your TM1 Server datasource URL such as *https://9.30.204.176:8010/api/v1/*. Then click *View Certificate* to open the server certification.
 - b. In the new Firefox browser window, click *PEM (cert)* to download the *tm1server.pem* file.

- From the command line, go to the SPSS Modeler jre\bin directory and then run the following command to import the tm1server.pem file to the SPSS Modeler Server (modify the paths to match your environment, as needed):

```
C:\Program Files\IBM\SPSS\Modeler\18.3\jre\bin>
keytool.exe -import -alias tm1server -file C:
\Users\Administrator\Downloads\tm1server.pem -keystore ..\lib\security\cacerts
```

- Enter the default keystore password **changeit** if prompted.
- When prompted whether to trust this certificate, enter **yes**.

- Restart SPSS Modeler Client and SPSS Modeler Server.

You can now use HTTPS and the SSL secured port number to connect to Cognos TM1.

Configuring the URL prefix

If IBM® SPSS® Collaboration and Deployment Services Repository is set up for SSL access, the value of the URL Prefix configuration setting must be modified as follows:

- Log in to the repository using browser-based console.

- Open *URL Prefix* configuration option.

Configuration > Setup > URL Prefix

- Set the value of the prefix to **https** instead of **http** and set the port value to the SSL port number. For example:

```
[default]
http://<hostname>:<port>
[SSL-enabled]
https://<hostname>:<SSLport>
```

Securing LDAP with SSL

Lightweight Directory Access Protocol (LDAP) is an Internet Engineering Task Force (IETF) standard for exchanging information between network directories and databases containing any level of information. For systems requiring additional security, LDAP providers, such as Microsoft's Active Directory, can operate over Secure Socket Layer (SSL), provided that the Web or application server supports LDAP over SSL. Using SSL in conjunction with LDAP can ensure that login passwords, application information, and other sensitive data are not hijacked, compromised, or stolen.

The following example illustrates how to enable LDAPS using Microsoft's Active Directory as a security provider. For more specific information on any of the steps or to find details that address a particular release of the security provider, see the original vendor documentation.

- Verify that Active Directory and the Enterprise Certificate Authority are installed and functioning.
- Use the certificate authority to generate a certificate, and import the certificate into the certificate store of the IBM® SPSS® Deployment Manager installation. This allows the LDAPS connection to be established between the IBM SPSS Collaboration and Deployment Services Repository and an Active Directory server.

To configure IBM SPSS Deployment Manager for secure Active Directory connections, verify that a connection exists to the repository.

- Launch the IBM SPSS Deployment Manager.
- From the Tools menu, choose Server Administration.
- Log in to a previously defined administered server.
- Double-click the Configuration icon for the server to expand the hierarchy.
- Double-click the Security Providers icon to expand the hierarchy.
- Double-click the Active Directory security provider.
- Enter configuration values for the instance of Active Directory with security certificates installed.
- Select the Use SSL check box.
- Note the name in the Domain User field. Subsequent logins using Active Directory are authenticated using SSL.

For additional information about installing, configuring, and implementing LDAPS on a particular application server, see the original vendor's documentation.

Related information

- [How SSL works](#)
- [Securing client/server and server-server communications with SSL](#)
- [Configuring the URL prefix](#)

Configuring groups

An authenticated user typically belongs to one or more security groups, and when group-based configuration is enabled for SPSS® Modeler Server, these groups can be used to permit or deny login to the server, or to customize the option settings for the user session.

Group configuration is supported in the following scenarios:

- In a *default* installation where the SPSS Modeler Server service runs under the Local System or root account and the user logs in with explicit credentials or using Single Sign-On (SSO): in this case the groups are the user's operating system security groups used to control file access, etc.
- In a *non-root* installation where the SPSS Modeler Server service runs under a non-privileged account and the user logs in using SSO: in this case the groups are the LDAP groups associated with the SSO principal. These groups are obtained from the LDAP security provider in IBM® SPSS Collaboration and Deployment Services, so some additional configuration is required to enable this scenario. See [Getting the SSO user's group membership](#) for more information.

If neither of these two scenarios apply, then the user's groups are not available and group configuration is not supported. In particular, in a *non-root* installation where the SPSS Modeler Server service runs under a non-privileged account and the user logs in using a user name and password, then the operating system groups are not available to the server and group configuration is not supported.

The principle of group-based configuration is that the option settings applied to a user's session can differ according to the user's group membership. These are the server-side settings normally read from the SPSS Modeler Server options.cfg file set identically for all sessions. The options.cfg file provides the defaults for all sessions, but there can be group-specific configuration files that override a subset of settings for particular sessions.

Group configuration allows for control of various settings, such as:

- Controlling file and DSN access
- Controlling resource usage

When the **group_configuration** option is enabled in options.cfg, IBM SPSS Modeler Server checks the groups.cfg file which controls who can log on to the server. The default is **N**. Following is a groups.cfg example that denies the **Test** group access to the server and allows the **Fraud** group access with a specified configuration. The asterisk allows all other groups access with the default configuration.

```
Test, DENY
Fraud, "groups/fraud.cfg"
*,
```

A specific group configuration, such as that for Fraud above, might restrict access to particular data sources or change resource settings (relating to SQL push back, memory usage, multi-threading, etc.) to enhance performance for members of that group.

The group configuration mechanism is designed to answer two questions:

1. Is the user allowed to use this instance of IBM SPSS Modeler Server?
2. If they are so allowed, then what configuration options do they get?

Regarding #2, the configuration options are those defined by options.cfg, and the default configuration refers to the settings in that file. The group mechanism allows you to override some of the default settings by specifying alternative configuration files for certain groups, where the settings in the group files take precedence over the defaults. The following parameters are supported in group configuration files. Note that there may be other parameters not listed here that can also be used in group configuration files.

```
sql_generation_enabled
db_udf_enabled
stream_rewriting_enabled
io_buffer_size
max_file_size
max_transfer_size
max_sql_string_length
default_sql_string_length
data_files_restricted
data_file_path
program_files_restricted
program_file_path
allow_modelling_memory_override
modelling_memory_limit_percentage
max_parallelism
enable_database_caching
sql_row_array_size
sql_data_sources_restricted
sql_data_source_path
memory_usage
sql_generation
sql_logging
sql_generation_logging
sql_log_native
```

```
sql_log_prettyprint
stream_rewriting
stream_rewriting_maximise_sql
date_baseline
date_2digit_baseline
time_rollover
date_format
time_format
decimal_separator
angles_in_radians
record_count_feedback_interval
record_count_suppress_input
decimal_places
column_width
cache_compression
enable_parallelism
database_caching
shell
use_bigint_for_count
trace_extension
```

Regarding #1, when group configuration is disabled, then everyone is allowed to use the server. When group configuration is enabled, then nobody is allowed to use it unless they are explicitly granted access in groups.cfg. So an empty groups.cfg file makes the server unusable by all. Typically, you add to groups.cfg the groups who should have access. For example:

```
A,
B,
C,
```

Optionally, for any group that you allow access, you can also specify a configuration file that overrides the default settings from options.cfg:

```
A, "a.cfg"
B, "b.cfg"
C, "c.cfg"
```

Any group for which you don't specify a configuration uses the default configuration, which comprises the settings from options.cfg.

The **DENY** option is allowed for more complex cases where a simple enumeration would grant more access than you really want. For example, you allocate a service for Fraud, but there are some developers who are also in the Fraud group who shouldn't have access. So you write:

```
devops, DENY
fraud,
```

You don't need to specify a default **DENY** because everyone else is excluded by virtue of not being included.

Note that this mechanism is subsidiary to the O/S logon mechanism (LDAP, etc). The user must always log on first, and if the O/S denies them access then they never get this far. If they can log on, then their O/S group membership is used to determine the group configuration, and they may be denied access at that point.

Controlling DSN access by group

Multi-factor authentication (MFA) requires that users can be restricted in the set of ODBC data source names (DSNs) that they are allowed to access according to their group membership.

The scheme for accomplishing this is similar to the existing scheme for file access. Two configuration settings are available in options.cfg:

```
sql_data_sources_restricted, N
sql_data_source_path, ""
```

If **sql_data_sources_restricted** is set to **Y**, then the user is limited to the DSNs listed in the associated path. DSNs are separated by the standard path separator character ; (semi-colon) on Windows and : (colon) on UNIX. For example, on Windows:

```
sql_data_sources_restricted, Y
sql_data_source_path, "Fraud - Analytic;Fraud - Operational"
```

When this restriction is enabled, it has the following results:

- When a user browses for data sources (for example, from the ODBC connection dialog, or when using the PSAPI Session `getServerDataSourceNames` API), instead of being presented with all the DSNs defined on the server system, the user will only see the subset of DSNs that is defined in the options.cfg path. Note that the path may contain DSNs that are not defined on the server, and these are ignored -- the user will not see those names.
- If a user constructs an ODBC node (or any node using an ODBC connection) that uses a script or PSAPI and the user specifies a DSN that is not included in the options.cfg path, the node will not run and the user will be presented with an error similar to Access denied to data source: <X>.

The data source path can include the **PATH**, **GROUP** and **USER** insertions described elsewhere for file paths. The **PATH** insertion allows the path to be constructed incrementally according to the user's group membership when group-based configuration is used. There might also be

circumstances where it makes sense to name a DSN after the group which owns it.

Building on the previous example, if access to the **Fraud** data sources is allowed only for members of the Fraud Analysts group, then the site can enable group configuration and create a configuration specific to the Fraud Analysts containing at least this line:

```
sql_data_source_path, "${PATH};Fraud - Analytic;Fraud - Operational"
```

The addition of the **PATH** prefix in this example ensures that the fraud analysts are still able to access other data sources allowed to everyone, or to other groups of which they are members.

Log files

IBM® SPSS® Modeler Server keeps a record of its important actions in a log file called `server_logging.log`. On UNIX, this file is in the `log` folder in the installation directory. On Windows, this file is in: `%ALLUSERSPROFILE%/IBM/SPSS/Modeler Server/<version>/log`.

The settings that control how logging is carried out in your installation are contained in the `log4cxx.properties` file.

Change the location of the log file

The default location of the log file is set, in the `log4cxx.properties` file, as:

```
log4j.appenders.MainLog.File=${app_log_location}/${PROFILE_NAME}/${app_type}logging.log
```

To change the log file location, edit this entry.

Enable tracing

There may be occasions when you require finer detail than just a basic list of information that shows the main actions; for example, this detail might be asked for by your support staff to help identify an issue. In these situations, you can amend the log to provide more detailed trace information.

To enable tracing, in the `log4cxx.properties` file, disable the line `log4j.rootLogger=INFO, MainLog, ConsoleLog` and enable the following line in its place: `log4j.rootLogger=TRACE, MainLog, TraceLog`

To change the location of the trace log, edit the entry:

```
log4j.appenders.TraceLog.File=${app_log_location}/${PROFILE_NAME}/${app_type}tracing_${PROCESS_ID}.log
```

Amend logging options

The `log4cxx.properties` file contains the controls that define how various events are logged. These controls are normally set to either **INFO** to record actions in the log file, or **WARN** to notify the user of a potential problem. If you are using the log file to identify potential errors, you can also set some of the controls to **TRACE**.

Control the size of the log file

By default, the log file continues to grow in size every time you use SPSS Modeler Server. To prevent the log becoming too large, you can either set it to be started from scratch every day, or define a size limit for it.

To set the log to be started as a new log every day, in the `log4cxx.properties` file, use the following entries:

```
log4j.appenders.MainLog=org.apache.log4j.DailyRollingFileAppender  
log4j.appenders.MainLog.DatePattern=''.yyyy-MM-dd
```

Alternatively, to define a size limit for the log (for example 8 Mb), in the `log4cxx.properties` file, use the following entries:

```
log4j.appenders.MainLog=org.apache.log4j.RollingFileAppender  
log4j.appenders.MainLog.MaxFileSize=8MB
```

Client log file

Note that you can also enable logging for IBM SPSS Modeler client. To do so, open the file `log4j2.xml` in a text editor and change `level="info"` to `level="debug"` in this line:

```
<Logger name="com.spss" additivity="false" level="info">
```

On Mac, the default client log file location is /Applications/IBM/SPSS/Modeler/18.4/Resources/log/. On Windows, the default location is \${env.USERPROFILE}/BM/SPSS/Modeler/18.4/log, where env.USERPROFILE is usually C:\Users%username% with %username% being the proper folder name.

Performance Overview

Real performance in analyzing data is affected by a number of factors, from server and database configuration to the ordering of individual nodes within a stream. In general, you can obtain the best performance by doing the following:

- Store your data in a DBMS, and use SQL generation and optimization whenever possible.
- Use hardware that meets or exceeds the recommendations given in [Architecture and Hardware Recommendations](#).
- Ensure that the client and server performance and optimization settings are properly configured. Note that when SPSS® Modeler is connected to an SPSS Modeler Server installation, the server performance and optimization settings override the client equivalents.
- Design streams for maximum performance.

More information about each of these performance factors is available in the following sections.

- [Server performance and optimization settings](#)
- [Client Performance and Optimization Settings](#)
- [Database Usage and Optimization](#)
- [Stream performance](#)

Many factors can impact how your SPSS Modeler streams perform.

Server performance and optimization settings

Certain IBM® SPSS® Modeler Server settings can be configured to optimize performance. You can adjust these settings using the IBM SPSS Modeler Administration Console interface included in IBM SPSS Deployment Manager. See the topic [IBM SPSS Modeler Server Administration](#) for more information.

The settings are grouped under the heading Performance and Optimization in the IBM SPSS Modeler Administration Console configuration window. The settings are preconfigured for optimal performance for most installations. However, you may need to adjust them depending on your particular hardware, the size of your data sets, and the contents of your streams. See the topic [Performance/Optimization](#) for more information.

Related information

- [IBM SPSS Modeler Server Administration](#)
- [Performance/Optimization](#)

Client Performance and Optimization Settings

The client performance and optimization settings are available from the Options tab of the Stream Properties dialog box. To display these options, choose the following from the client menu.

Tools > Stream Properties > Options > Optimization

You can use the Optimization settings to optimize stream performance. Note that the performance and optimization settings on IBM® SPSS® Modeler Server (if used) override any equivalent settings in the client. If these settings are disabled in the server, then the client cannot enable them. But if they are enabled in the server, the client can choose to disable them.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > About > Additional Details

If connectivity is enabled, you see the option Server Enablement in the License Status tab.

See [Connecting to IBM SPSS Modeler Server](#) for more information.

Note: Whether SQL pushback and optimization are supported depends on the type of database in use. For the latest information on which databases and ODBC drivers are supported and tested for use with IBM SPSS Modeler, see the corporate Support site at <http://www.ibm.com/support>.

Enable stream rewriting. Select this option to enable stream rewriting in IBM SPSS Modeler. Four types of rewriting are available, and you can select one or more of them. Stream rewriting reorders the nodes in a stream behind the scenes for more efficient operation, without altering stream semantics.

- Optimize SQL generation. This option enables nodes to be reordered within the stream so that more operations can be pushed back using SQL generation for execution in the database. When it finds a node that cannot be rendered into SQL, the optimizer will look ahead to see if there are any downstream nodes that can be rendered into SQL and safely moved in front of the problem node without affecting the stream semantics. Not only can the database perform operations more efficiently than IBM SPSS Modeler, but such pushbacks act to reduce the size of the data set that is returned to IBM SPSS Modeler for processing. This, in turn, can reduce network traffic and speed stream operations. Note that the Generate SQL check box must be selected for SQL optimization to have any effect.
- Optimize CLEM expression. This option enables the optimizer to search for CLEM expressions that can be preprocessed before the stream is run, in order to increase the processing speed. As a simple example, if you have an expression such as $\log(\text{salary})$, the optimizer would calculate the actual salary value and pass that on for processing. This can be used both to improve SQL pushback and IBM SPSS Modeler Server performance.
- Optimize syntax execution. This method of stream rewriting increases the efficiency of operations that incorporate more than one node containing IBM SPSS Statistics syntax. Optimization is achieved by combining the syntax commands into a single operation, instead of running each as a separate operation.
- Optimize other execution. This method of stream rewriting increases the efficiency of operations that cannot be delegated to the database. Optimization is achieved by reducing the amount of data in the stream as early as possible. While maintaining data integrity, the stream is rewritten to push operations closer to the data source, thus reducing data downstream for costly operations, such as joins.

Enable parallel processing. When running on a computer with multiple processors, this option allows the system to balance the load across those processors, which may result in faster performance. Use of multiple nodes or use of the following individual nodes may benefit from parallel processing: C5.0, Merge (by key), Sort, Bin (rank and tile methods), and Aggregate (using one or more key fields).

Generate SQL. Select this option to enable SQL generation, allowing stream operations to be pushed back to the database by using SQL code to generate execution processes, which may improve performance. To further improve performance, Optimize SQL generation can also be selected to maximize the number of operations pushed back to the database. When operations for a node have been pushed back to the database, the node will be highlighted in purple when the stream is run.

- Database caching. For streams that generate SQL to be executed in the database, data can be cached midstream to a temporary table in the database rather than to the file system. When combined with SQL optimization, this may result in significant gains in performance. For example, the output from a stream that merges multiple tables to create a data mining view may be cached and reused as needed. With database caching enabled, simply right-click any nonterminal node to cache data at that point, and the cache is automatically created directly in the database the next time the stream is run. This allows SQL to be generated for downstream nodes, further improving performance. Alternatively, this option can be disabled if needed, such as when policies or permissions preclude data being written to the database. If database caching or SQL optimization is not enabled, the cache will be written to the file system instead. See the topic [Caching options for nodes](#) for more information.
- Use relaxed conversion. This option enables the conversion of data from either strings to numbers, or numbers to strings, if stored in a suitable format. For example, if the data is kept in the database as a string, but actually contains a meaningful number, the data can be converted for use when the pushback occurs.

Note: Due to minor differences in SQL implementation, streams run in a database may return slightly different results from those returned when run in IBM SPSS Modeler. For similar reasons, these differences may also vary depending on the database vendor.

Database Usage and Optimization

Database server. If possible, create a dedicated database instance for data mining so that the production server is not impacted by IBM® SPSS® Modeler queries. SQL statements generated by IBM SPSS Modeler can be demanding—multiple tasks on the IBM SPSS Modeler Server machine can be executing SQL in the same database.

In-database mining. Many database vendors provide data mining extensions for their products. These extensions allow data mining activities (such as model-building or scoring) to run within the database server, or within a separate dedicated server. IBM SPSS Modeler's in-database mining features complement and extend its SQL generation capability, providing a way to drive the vendor-specific database extensions. In some cases, taking this approach avoids the potentially expensive overhead of data transfer between IBM SPSS Modeler and the database. Database caching can further increase the benefits. For more information, see the file *DatabaseMiningGuide.pdf*, which is available as a part of your downloaded eImage.

- [SQL Optimization](#)

SQL Optimization

For best performance, you should always try to maximize the amount of SQL generated to exploit the performance and scalability of the database. Only the parts of the stream that cannot be compiled to SQL should be executed within IBM® SPSS® Modeler Server. For more information, see [SQL optimization](#).

Uploading File-Based Data

Data that is not stored in a database cannot benefit from SQL optimization. If the data you want to analyze is not already in a database, you can upload it using a Database Output node. You can also use this node to store intermediate data sets from data preparation and the results of deployment.

IBM SPSS Modeler can interface with the external loaders for many common database systems. Several scripts are included with the software and are available (with documentation) in the `/scripts` subdirectory under your IBM SPSS Modeler installation folder.

The following table shows the potential performance benefit of bulk-loading. The figures show the elapsed time to export 250,000 records and 21 fields to an Oracle database. The external loader was Oracle's `sqlldr` utility.

Table 1. Performance benefit of bulk-loading

Export option	Time (in seconds)
Default (ODBC)	409
Bulk-load via ODBC	52
Bulk-load via external loader	33

Stream performance

Many factors can impact how your SPSS® Modeler streams perform.

Keep these general tips in mind:

- Where possible, consider minimizing the size of your data by limiting processing to only those fields that are needed by using Filter nodes and the Filter tab in source nodes.
- Leverage in-database processing capability whenever possible to reduce the amount of data pulled in to SPSS Modeler.
- Minimize the network distance between your IBM® SPSS Modeler Server and the source data.
- Certain data sources require more overhead than others. For example, the Excel source node takes longer to access the same data than a CSV file. XML data is inherently wasteful and shouldn't be used for storing large amounts of data.
- If using Python-based nodes or R-based nodes, note that there are internal data transfers that must take place. This can sometimes slow processing.
- Accomplishing your tasks with the fewest number of nodes is usually preferable to more nodes.
- Use Type nodes only when necessary. This is especially true when Hadoop is the data source because each Type node processes the entire data flow. See [What is instantiation?](#).
- Certain statistical modeling nodes might be slow, especially with data sets that have many categorical fields.
- Changing the order of nodes can influence processing speed, so experiment with node order. For example, if you have a stream with nodes that reduce data by subsetting or reducing the number of fields, move them as early in the stream as possible.
- If a modeling node you're using has a corresponding -AS version, use the -AS node instead because it's multi-threaded and can improve processing.

SQL optimization

One of the most powerful capabilities of IBM® SPSS® Modeler is the ability to perform many data preparation and mining operations directly in the database. By generating SQL code that can be pushed back to the database for execution, many operations, such as sampling, sorting, deriving new fields, and certain types of graphing, can be performed in the database rather than on the IBM SPSS Modeler or IBM SPSS Modeler Server computer. When you are working with large data sets, these *pushbacks* can dramatically enhance performance in several ways:

- By reducing the size of the result set to be transferred from the DBMS to IBM SPSS Modeler. When large result sets are read through an ODBC driver, network I/O or driver inefficiencies can result. For this reason, the operations that benefit most from SQL optimization are row and column selection and aggregation (Select, Sample, Aggregate nodes). These operations typically reduce the size of the data set to be transferred. Data can also be cached to a temporary table in the database at critical points in the stream (after a Merge or Select node, for example) to further improve performance.
- By using the performance and scalability of the database. Efficiency is increased because a DBMS can often take advantage of parallel processing, more powerful hardware, more sophisticated management of disk storage, and the presence of indexes.

Given these advantages, IBM SPSS Modeler is designed to maximize the amount of SQL generated by each stream so that only those operations that can't be compiled to SQL will be run by IBM SPSS Modeler Server. Because of limitations in what can be expressed in standard SQL (SQL-92), however, certain operations may not be supported. For more information, see [Tips for maximizing SQL generation](#).

Note: Keep the following information in mind when working with SQL:

- Because of minor differences in SQL implementation, streams that run in a database can return slightly different results when they run in IBM SPSS Modeler. These differences may also vary depending on the database vendor, for similar reasons. For example, depending on the database configuration for case sensitivity in string comparison and string collation, IBM SPSS Modeler streams that run by using SQL pushback might produce different results from those that run without SQL pushback. Contact your database administrator for advice on configuring your database. To maximize compatibility with IBM SPSS Modeler, we recommend making sure database string comparisons are case-sensitive.
- Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity is enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, from the IBM SPSS Modeler menu, go to:
 - Help > About > Additional Details

If connectivity is enabled, you see the option Server Enablement in the License Status tab.

See [Connecting to IBM SPSS Modeler Server](#) for more information.

- When using IBM SPSS Modeler to generate SQL, the result that uses SQL push back might not be consistent with IBM SPSS Modeler native on some platforms (Linux/zLinux, for example). The reason is that floating point is handled differently on different platforms.

Note: When you run streams in a Netezza database, date and time details are taken from that database. This behavior might differ from your local or IBM SPSS Modeler Server date and time if, for example, the database is on a machine that is located in a different country or time zone.

Database requirements

For more information about which databases and ODBC drivers are supported and tested for use with IBM SPSS Modeler, see the product compatibility matrices on the corporate Support site at <http://www.ibm.com/support>.

You might gain more performance improvements by using database modeling.

ODBC driver setup

To ensure that time details (such as HH:MM:SS) are processed correctly when using SQL 2012 on Windows 32bit systems, when setting up your ODBC SQL Server Wire Protocol Driver, select both the Enable Quoted Identifiers and Fetch TWFS as Time options.

- [How SQL generation works](#)
- [Configuring SQL Optimization](#)
- [Previewing Generated SQL](#)
- [Viewing SQL for Model Nuggets](#)
- [Tips for maximizing SQL generation](#)
- [Nodes supporting SQL generation](#)
- [CLEM Expressions and Operators Supporting SQL Generation](#)
- [Writing SQL Queries](#)
- [Scoring adapter for Teradata - duplicate rows](#)

How SQL generation works

The initial fragments of a stream leading from the database source nodes are the main targets for SQL generation. When a node is encountered that cannot be compiled to SQL, the data are extracted from the database and subsequent processing is performed by IBM® SPSS® Modeler Server.

During stream preparation and prior to execution, the SQL generation process happens as follows:

- The server reorders streams to move downstream nodes into the “SQL zone” where it can be proven safe to do so. (This feature can be disabled on the server.)
- Working from the source nodes toward the terminal nodes, SQL expressions are constructed incrementally. This phase stops when a node is encountered that cannot be converted to SQL or when the terminal node (for example, Table node or Graph node) is converted to SQL. At the end of this phase, each node is labeled with an SQL statement if the node and its predecessors have an SQL equivalent.
- Working from the nodes with the most complicated SQL equivalents back toward the source nodes, the SQL is checked for validity. The SQL that was successfully validated is chosen for execution.
- Nodes for which all operations have generated SQL are highlighted in purple on the stream canvas. Based on the results, you may want to further reorganize your stream where appropriate to take full advantage of database execution. See the topic [Tips for maximizing SQL generation](#) for more information.

Where Improvements Occur

SQL optimization improves performance in a number of data operations:

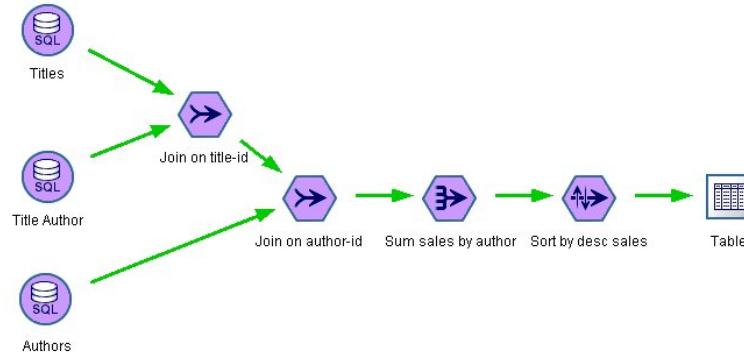
- Joins (merge by key). Join operations can increase optimization within databases.
- Aggregation. The Aggregate, Distribution, and Web nodes all use aggregation to produce their results. Summarized data uses considerably less bandwidth than the original data.
- Selection. Choosing records based on certain criteria reduces the quantity of records.
- Sorting. Sorting records is a resource-intensive activity that is performed more efficiently in a database.
- Field derivation. New fields are generated more efficiently in a database.
- Field projection. IBM SPSS Modeler Server extracts only fields that are required for subsequent processing from the database, which minimizes bandwidth and memory requirements. The same is also true for superfluous fields in flat files: although the server must read the superfluous fields, it does not allocate any storage for them.
- Scoring. SQL can be generated from decision trees, rulesets, linear regression, and factor-generated models.

• [SQL Generation Example](#)

SQL Generation Example

The following stream joins three database tables by key operations and then performs an aggregation and sort.

Figure 1. Optimized stream with purple nodes indicating SQL pushbacks (operations performed in database)



Generated SQL

The generated SQL for this stream is as follows:

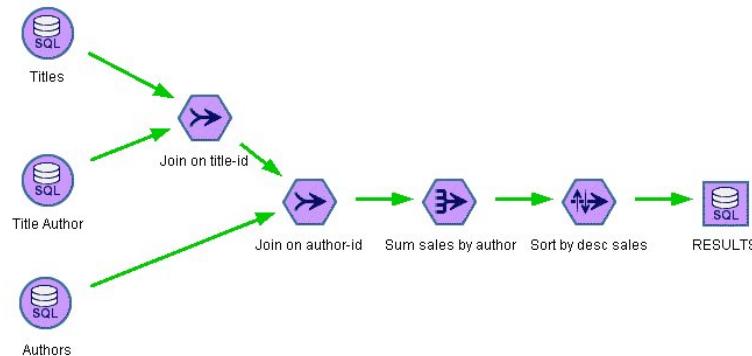
```

SELECT
    T2. au_lname AS C0,
    T2. au_fname AS C1,
    SUM({fn CONVERT(T0. ytd_sales ,SQL_BIGINT)}) AS C2
FROM
    dbo . titles T0,
    dbo . titleauthor T1,
    dbo . authors T2
WHERE
    (T0. title_id = T1. title_id )
    AND (T1. au_id = T2. au_id )
GROUP BY T2. au_lname ,T2. au_fname
ORDER BY 3 DESC
  
```

Executing the Stream

When the stream is terminated with a database export node, it is possible for the whole stream to be executed in the database.

Figure 2. Entire stream executed in database



Related information

- [SQL optimization](#)
 - [Configuring SQL Optimization](#)
 - [Tips for maximizing SQL generation](#)
 - [Nodes supporting SQL generation](#)
 - [CLEM Expressions and Operators Supporting SQL Generation](#)
 - [Configuring Oracle for SQL Optimization](#)
-

Configuring SQL Optimization

1. Install an ODBC driver and configure a data source for the database you want to use. See the topic [Data Access](#) for more information.
2. Create a stream that uses a source node to pull data from that database.
3. Check to be sure that SQL generation is enabled on the client and server, if applicable. It is enabled by default for both.

To Enable SQL Optimization on the Client

1. From the Tools menu, choose Stream Properties...> Options.
2. Click the Optimization tab. Select Generate SQL to enable SQL optimization. Optionally, you can select other settings to improve performance. See the topic [Client Performance and Optimization Settings](#) for more information.

To Enable SQL Optimization on the Server

Because server settings override any specifications made on the client, the server configuration settings Stream Rewriting and Automatic SQL Generation must both be turned on. For more information about how to change IBM® SPSS® Modeler Server settings, see the section [Performance/Optimization](#). Note that if these settings are disabled in the server, then the client cannot enable them. But if they are enabled in the server, the client can choose to disable them.

To Enable Optimization When Scoring Models

For purposes of scoring, SQL generation must be enabled separately for each modeling node, regardless of any server or client-level settings. This is done because some models generate extremely complex SQL expressions that may not be evaluated effectively within the database. The database may report errors when trying to execute the generated SQL, due to the size or complexity of the SQL.

A certain amount of trial-and-error may be needed to determine whether SQL generation improves performance for a given model. This is done on the Settings tab after adding a generated model to a stream.

Related information

- [How SQL generation works](#)
 - [SQL Generation Example](#)
 - [Tips for maximizing SQL generation](#)
 - [Nodes supporting SQL generation](#)
 - [CLEM Expressions and Operators Supporting SQL Generation](#)
 - [Configuring Oracle for SQL Optimization](#)
-

Previewing Generated SQL

You can preview generated SQL in the message log before executing it in the database. This may be useful for debugging purposes, and it allows you to export the generated SQL to edit or run in the database at a future date. It also indicates which nodes will be pushed back to the database, which may help you determine whether the stream can be reordered to improve performance.

1. Make sure that Display SQL in the messages log during stream execution and Display SQL generation details in the messages log during stream preparation are selected in the User Options dialog box. See the topic [Client Performance and Optimization Settings](#) for more information.
2. On the stream canvas, select the node or stream that you want to preview.
3. Click the Preview SQL button on the toolbar.
All nodes for which SQL is generated (and that will be pushed back to the database when the stream is executed) are colored purple on the stream canvas.
4. To preview the generated SQL, from the menus choose:
Tools...>Stream Properties...>Messages...

Related information

- [Writing SQL Queries](#)
 - [Configuring Oracle for SQL Optimization](#)
-

Viewing SQL for Model Nuggets

For some models, SQL for the model nugget can be generated, pushing back the model scoring stage to the database. The main use of this feature is not to improve performance, but to allow streams containing these nuggets to have their full SQL pushed back. See the topic [Nodes supporting SQL generation](#) for more information.

To view the SQL for a model nugget that supports SQL generation:

1. Select the Settings tab on the model nugget.
 2. Choose one of the options Generate with (no) missing value support or Generate SQL for this model, as appropriate.
 3. In the model nugget menu, choose:
File > Export SQL
 4. Save the file.
 5. Open the file to view the SQL.
-

Tips for maximizing SQL generation

To get the best performance boost from SQL optimization, pay attention to the following items.

Stream order. SQL generation may be halted when the function of the node has no semantic equivalent in SQL because IBM® SPSS® Modeler's data-mining functionality is richer than the traditional data-processing operations supported by standard SQL. When this happens, SQL generation is also suppressed for any downstream nodes. Therefore, you may be able to significantly improve performance by reordering nodes to put operations that halt SQL as far downstream as possible. The SQL optimizer can do a certain amount of reordering automatically (just make sure stream rewriting is enabled), but further improvements may be possible. A good candidate for this is the Select node, which can often be brought forward. See the topic [Nodes supporting SQL generation](#) for more information.

CLEM expressions. If a stream cannot be reordered, you may be able to change node options or CLEM expressions or otherwise recast the way the operation is performed, so that it no longer inhibits SQL generation. Derive, Select, and similar nodes can commonly be rendered into SQL, provided that all of the CLEM expression operators have SQL equivalents. Most operators can be rendered, but there are a number of operators that inhibit SQL generation (in particular, the sequence functions ["@ functions"]). Sometimes generation is halted because the generated query has become too complex for the database to handle. See the topic [CLEM Expressions and Operators Supporting SQL Generation](#) for more information.

Multiple source nodes. Where a stream has multiple database source nodes, SQL generation is applied to each input branch independently. If generation is halted on one branch, it can continue on another. Where two branches merge (and both branches can be expressed in SQL up to the merge), the merge itself can often be replaced with a database join, and generation can be continued downstream.

Database algorithms. Model estimation is always performed on IBM SPSS Modeler Server rather than the database, except when using database-native algorithms from Microsoft, IBM, or Oracle.

Scoring models. In-database scoring is supported for some models by rendering the generated model into SQL. However, some models generate extremely complex SQL expressions that aren't always evaluated effectively within the database. For this reason, SQL generation must be enabled separately for each model node. If you find that a model node is inhibiting SQL generation, go to the Settings tab on the node dialog box and select Generate SQL for this model (with some models, you may have additional options controlling generation). Run tests to confirm that the option is beneficial for your application. See the topic [Nodes supporting SQL generation](#) for more information.

When testing modeling nodes to see if SQL generation for models works effectively, we recommend first saving all streams from IBM SPSS Modeler. Some database systems may hang while trying to process the (potentially complex) generated SQL, requiring IBM SPSS Modeler to be closed from the Windows task manager.

Database caching. If you are using a node cache to save data at critical points in the stream (for example, following a Merge or Aggregate node), make sure that database caching is enabled along with SQL optimization. This will allow data to be cached to a temporary table in the database (rather than the file system) in most cases. See the topic [Configuring SQL Optimization](#) for more information.

Vendor-specific SQL. Most of the generated SQL is standards-conforming (SQL-92), but some nonstandard, vendor-specific features are exploited where practical. The degree of SQL optimization can vary, depending on the database source.

SQL configuration option. By default, SPSS Modeler considers SQL queries written in an ODBC source node to be non-replayable, meaning that the query is considered to return different results when being executed multiple times. However, in some cases, this may prevent SPSS Modeler

from generating SQL for downstream nodes. You can override this behavior by changing the following value to `Y` in the `odbc-db2-custom-properties.cfg`. The file is located in the SPSS Modeler config directory.

```
assume_custom_sql_replayable, Y
```

Nodes supporting SQL generation

The following tables show nodes representing data-mining operations that support SQL generation. With the exception of the database modeling nodes, if a node does not appear in these tables, it does not support SQL generation.

You can preview the SQL that is generated before executing it. See the topic [Previewing Generated SQL](#) for more information.

Table 1. Sources

Node supporting SQL generation	Notes
Database	This node is used to specify tables and views to be used for further analysis. This node enables entry of SQL queries. Avoid results sets with duplicate column names. See the topic Writing SQL Queries for more information.

Table 2. Record operations

Node supporting SQL generation	Notes
Select	Supports generation only if SQL generation for the select expression itself is supported (see expressions below). If any fields have nulls, SQL generation does not give the same results for discard as are given in native IBM® SPSS® Modeler.
Sample	Simple sampling supports SQL generation to varying degrees depending on the database. See Table 3 .
Aggregate	SQL generation support for aggregation depends on the data storage type. See Table 4 .
RFM Aggregate	Supports generation except if saving the date of the second or third most recent transactions, or if only including recent transactions. However, including recent transactions does work if the <code>datetime_date(YEAR,MONTH,DAY)</code> function is pushed back.
Sort	
Merge	No SQL generated for merge by order. Merge by key with full or partial outer join is only supported if the database/driver supports it. Non-matching input fields can be renamed by means of a Filter node, or the Filter tab of a source node. Supports SQL generation for merge by condition. For all types of merge, <code>SQL_SP_EXISTS</code> is not supported if inputs originate in different databases.
Append	Supports generation if inputs are unsorted. Note: SQL optimization is only possible when your inputs have the same number of columns.
Distinct	A Distinct node with the (default) mode Create a composite record for each group selected does not support SQL optimization.

Table 3. SQL generation support in the Sample node for simple sampling

Mode	Sample	Max size	Seed	Db2 for z/OS	Db2 for OS/400	Db2 for Win/UNIX	Netezza	Oracle	SQL Server	Teradata
Include	First	n/a		Y	Y	Y	Y	Y	Y	Y
	1-in-n	off		Y	Y	Y	Y	Y		Y
		max		Y	Y	Y	Y	Y		Y
	Random %	off	off	Y		Y	Y	Y		Y
		on	Y			Y		Y		
		max	off	Y		Y	Y	Y		Y
			on	Y		Y		Y		
Discard	First	off					Y	Y		
		max					Y	Y		
	1-in-n	off		Y	Y	Y	Y	Y		Y
		max		Y	Y	Y	Y	Y		Y
	Random %	off	off	Y		Y	Y	Y		Y
		on	Y			Y		Y		
		max	off	Y		Y	Y	Y		Y
			on	Y		Y		Y		

Table 4. SQL generation support in the Aggregate node

Storage	Sum	Mean	Min	Max	SDev	Median	Count	Variance	Percentile
Integer	Y	Y	Y	Y	Y	Y*	Y	Y	Y*

Storage	Sum	Mean	Min	Max	SDev	Median	Count	Variance	Percentile
Real	Y	Y	Y	Y	Y	Y*	Y	Y	Y*
Date			Y	Y		Y*	Y		Y*
Time			Y	Y		Y*	Y		Y*
Timestamp			Y	Y		Y*	Y		Y*
String			Y	Y		Y*	Y		Y*

* Median and Percentile are supported on Oracle.

Table 5. Field operations

Node supporting SQL generation	Notes
Type	Supports SQL generation if the Type node is instantiated and no ABORT or WARN type checking is specified.
Filter	
Derive	Supports SQL generation if SQL generated for the derive expression is supported (see expressions below).
Ensemble	Supports SQL generation for Continuous targets. For other targets, supports generation only if the "Highest confidence wins" ensemble method is used.
Filler	Supports SQL generation if the SQL generated for the derive expression is supported (see expressions below).
Anonymize	Supports SQL generation for Continuous targets, and partial SQL generation for Nominal and Flag targets.
Reclassify	
Binning	Supports SQL generation if the "Tiles (equal count)" binning method is used and the "Read from Bin Values tab if available" option is selected. Note: Due to differences in the way that bin boundaries are calculated (this is caused by the nature of the distribution of data in bin fields), you might see differences in the binning output when comparing normal stream execution results and SQL pushback results. To avoid this, use the Record count tiling method, and either Add to next or Keep in current tiles to obtain the closest match between the two methods of stream execution.
RFM Analysis	Supports SQL generation if the "Read from Bin Values tab if available" option is selected, but downstream nodes will not support it.
Partition	Supports SQL generation to assign records to partitions.
SetToFlag	
Restructure	

Table 6. Graphs

Node supporting SQL generation	Notes
Graphboard	SQL generation is supported for the following graph types: Area, 3-D Area, Bar, 3-D Bar, Bar of Counts, Heat map, Pie, 3-D Pie, Pie of Counts. For Histograms, SQL generation is supported for categorical data only. SQL generation is not supported for Animation in Graphboard.
Distribution	
Web	
Evaluation	

For some models, SQL for the model nugget can be generated, pushing back the model scoring stage to the database. The main use of this feature is not to improve performance, but to allow streams containing these nuggets to have their full SQL pushed back. See the topic [Viewing SQL for Model Nuggets](#) for more information.

Table 7. Model nuggets

Model nugget supporting SQL generation	Notes
C&R Tree	Supports SQL generation for the single tree option, but not for the boosting, bagging or large dataset options.
QUEST	
CHAID	
C5.0	
Decision List	
Linear	Supports SQL generation for the standard model option, but not for the boosting, bagging or large dataset options.
Neural Net	Supports SQL generation for the standard model option (Multilayer Perceptron only), but not for the boosting, bagging or large dataset options.
PCA/Factor	
Logistic	Supports SQL generation for Multinomial procedure but not Binomial. For Multinomial, generation is not supported when confidences are selected, unless the target type is Flag.
Generated Rulesets	

Model nugget supporting SQL generation	Notes
Auto Classifier	If a User Defined Function (UDF) scoring adapter is enabled, these nuggets support SQL pushback. In addition, If either SQL generation for Continuous targets, or the "Highest confidence wins" ensemble method are used, these nuggets support further pushback downstream.
Auto Numeric	

Table 8. Output

Node supporting SQL generation	Notes
Table	Supports generation if SQL generation is supported for highlight expression (see expressions below).
Matrix	Supports generation except if "All numerics" is selected for the Fields option.
Analysis	Supports generation, depending on the options selected.
Transform	
Graphs	SQL Pushback is not supported if animation is enabled in a Graphboard node.
Statistics	Supports generation if the Correlate option is not used.
Report	
Set Globals	

Table 9. Export

Node supporting SQL generation	Notes
Database	
Publisher	The published stream will contain generated SQL.

CLEM Expressions and Operators Supporting SQL Generation

The following tables show the mathematical operations and expressions that support SQL generation and are often used during data mining. Operations absent from these tables do not support SQL generation in the current release.

Table 1. Operators

Operation supporting SQL generation	Notes
+	
-	
/	
*	
><	Used to concatenate strings.

Table 2. Relational operators

Operation supporting SQL generation	Notes
=	
/=	Used to specify "not equal."
>	
>=	
<	
<=	

Table 3. Functions

Operation supporting SQL generation	Notes
abs	
allbutfirst	
allbutlast	
and	
arccos	
arcsin	
arctan	
arctanh	
cos	
div	
exp	
fracof	
hasstartstring	
hassubstring	

Operation supporting SQL generation	Notes
integer	
intof	
isaplhacode	
islowercode	
isnumbercode	
issstartstring	
issubstring	
isuppercode	
last	
length	
locchar	
log	
log10	
lowertoupper	
max	
member	
min	
negate	
not	
number	
or	
pi	
real	
rem	
round	
sign	
sin	
sqrt	
string	
strmember	
subscrs	
substring	
substring_between	
uppertolower	
to_string	

Table 4. Special functions

Operation supporting SQL generation	Notes
@NULL	
@GLOBAL_AVE	The special global functions are used to retrieve global values computed by the Set Globals node.
@GLOBAL_SUM	
@GLOBAL_MAX	
@GLOBAL_MEAN	
@GLOBAL_MIN	
@GLOBALSDDEV	

Table 5. Aggregate functions

Operation supporting SQL generation	Notes
Sum	
Mean	
Min	
Max	
Count	
SDev	

- [Using SQL Functions in CLEM Expressions](#)

Related information

- [How SQL generation works](#)
 - [SQL Generation Example](#)
 - [Configuring SQL Optimization](#)
 - [Tips for maximizing SQL generation](#)
 - [Nodes supporting SQL generation](#)
 - [Configuring Oracle for SQL Optimization](#)
-

Using SQL Functions in CLEM Expressions

The `@SQLFN` function can be used to add named SQL functions within CLEM expressions, for purposes of database execution only. This can be useful in special cases where proprietary SQL or other vendor-specific customizations are required.

Use of this function is not covered by the standard IBM® SPSS® Modeler support agreement, since execution relies on external database components beyond the control of IBM Corp., but may be deployed in special cases, typically as part of a Services engagement. Contact <http://www.ibm.com/software/analytics/spss/services/> for more information if necessary.

Writing SQL Queries

When using the Database node, you should pay special attention to any SQL queries that result in a dataset with duplicate column names. These duplicate names often prevent SQL optimization for any downstream nodes.

IBM® SPSS® Modeler uses nested **SELECT** statements to push back SQL for streams that use an SQL query in the Database source node. In other words, the stream nests the query specified in the Database source node inside of one or more **SELECT** statements generated during the optimization of downstream nodes. Therefore, if the result set of a query contains duplicate column names, the statement cannot be nested by the RDBMS. Nesting difficulties occur most often during a table join where a column with the same name is selected in more than one of the joined tables. For example, consider this query in the source node:

```
SELECT e.ID, e.LAST_NAME, d.*  
FROM EMP e RIGHT OUTER JOIN  
DEPT d ON e.ID = d.ID;
```

The query will prevent subsequent SQL optimization, since this **SELECT** statement would result in a dataset with two columns called **ID**.

In order to allow full SQL optimization, you should be more explicit when writing SQL queries and specify column aliases when a situation with duplicate column names arises. The statement below illustrates a more explicit query:

```
SELECT e.ID AS ID1, e.LAST_NAME, d.*  
FROM EMP e RIGHT OUTER JOIN  
DEPT d ON e.ID = d.ID;
```

Related information

- [Previewing Generated SQL](#)
 - [Configuring Oracle for SQL Optimization](#)
-

Scoring adapter for Teradata - duplicate rows

The IBM® SPSS® Modeler Server Scoring Adapter for Teradata expects no identical rows in its input data. Teradata does not allow the existence of two identical rows in a table. However, duplicate rows can happen when joining tables, or when the user uses only part of the fields of a table as input. These duplicate rows will lead to an incorrect number of records after a cartesian join.

Related information

- [How SQL generation works](#)
- [SQL Generation Example](#)
- [Configuring SQL Optimization](#)
- [Tips for maximizing SQL generation](#)
- [CLEM Expressions and Operators Supporting SQL Generation](#)
- [Configuring Oracle for SQL Optimization](#)

Configuring Oracle for UNIX Platforms

- [Configuring Oracle for SQL Optimization](#)

Configuring Oracle for SQL Optimization

When running IBM® SPSS® Modeler Server on UNIX platforms and reading from an Oracle database, consider the following tips to ensure that generated SQL is being fully optimized within the database.

Proper Locale Specification

When running IBM SPSS Modeler Server in a locale other than that shipped with the Connect ODBC drivers, you should reconfigure the machine to enhance SQL optimization. Connect ODBC drivers ship only with the *en_US* locale files. Consequently, if the IBM SPSS Modeler Server machine is running in a different locale or if the shell in which IBM SPSS Modeler Server was started did not have the locale fully defined, generated SQL may not be fully optimized within Oracle. The reasons are as follows:

- IBM SPSS Modeler Server uses the ODBC locale files corresponding to the locale in which it is running to translate the codes returned from the database into text strings. It then uses these text strings to determine which database it is actually connecting to.
- If the locale (as returned to IBM SPSS Modeler Server by the system `$LANG` query) is not *en_US*, IBM SPSS Modeler cannot translate the codes it receives from the ODBC driver into text. In other words, an untranslated code, rather than the string *Oracle*, is returned to IBM SPSS Modeler Server at the start of a database connection. This means that IBM SPSS Modeler is unable to optimize streams for Oracle.

To check and reset locale specifications:

1. In a UNIX shell, run:

```
#locale
```

This will return the locale information for the shell. For example:

```
$ locale
LANG=en_US.ISO8859-15
LC_CTYPE="en_US.ISO8859-15"
LC_NUMERIC="en_US.ISO8859-15"
LC_TIME="en_US.ISO8859-15"
LC_COLLATE="en_US.ISO8859-15"
LC_MONETARY="en_US.ISO8859-15"
LC_MESSAGES="en_US.ISO8859-15"
LC_ALL=en_US.ISO8859-15
```

2. Change to your Connect ODBC/locale directory. (Here you will see a single directory, *en_US*.)

3. Create a soft link to this *en_US* directory, specifying the name of the locale setup in the shell. An example is as follows:

```
#ln -s en_US en_US.ISO8859-15
```

For a non-English locale, such as *fr_FR.ISO8859-1*, you should create the soft link as follows:

```
#ln -s en_US fr_FR.ISO8859-1
```

4. Once you have created the link, restart IBM SPSS Modeler Server from this same shell. (IBM SPSS Modeler Server receives its locale information from the shell from which it is started.)

Notes

When optimizing a UNIX machine for SQL pushbacks to Oracle, consider the following tips:

- The full locale must be specified. In the above example, you must create the link in the form `language_territory.code-page`. The existing *en_US* locale directory is not sufficient.
- To fully optimize in-database mining, both `LANG` and `LC_ALL` must be defined in the shell used to start IBM SPSS Modeler Server. `LANG` may be defined in the shell as you would any other environment variable before restarting IBM SPSS Modeler Server. For example, see the following definition:

```
#LANG=en_US.ISO8859-15; export LANG
```

- Each time you start IBM SPSS Modeler Server you will need to check that the shell locale information is fully specified and that the appropriate soft link exists in the ODBC/locale directory.

Related information

- [Previewing Generated SQL](#)

- [Writing SQL Queries](#)
 - [SQL optimization](#)
 - [How SQL generation works](#)
 - [SQL Generation Example](#)
 - [Configuring SQL Optimization](#)
 - [Tips for maximizing SQL generation](#)
 - [Nodes supporting SQL generation](#)
 - [CLEM Expressions and Operators Supporting SQL Generation](#)
-

Configuring UNIX Startup Scripts

- [Introduction](#)
 - [Scripts](#)
 - [Automatically Starting and Stopping IBM SPSS Modeler Server](#)
 - [Manually Starting and Stopping IBM SPSS Modeler Server](#)
 - [Editing Scripts](#)
 - [Controlling Permissions on File Creation](#)
 - [IBM SPSS Modeler Server and the data access pack](#)
-

Introduction

This appendix describes some of the scripts that ship with the UNIX versions of IBM® SPSS® Modeler Server, and it explains how to configure the scripts. Scripts are used to:

- Configure IBM SPSS Modeler Server to start automatically when the server computer is restarted.
 - Manually stop and restart IBM SPSS Modeler Server.
 - Change permissions on files created by IBM SPSS Modeler Server.
 - Configure IBM SPSS Modeler Server to work with the ODBC Connect drivers provided with IBM SPSS Modeler Server. See the topic [IBM SPSS Modeler Server and the data access pack](#) for more information.
-

Scripts

IBM® SPSS® Modeler Server uses several scripts, including:

- modelersrv.sh. The manual startup script for IBM SPSS Modeler Server is located in the IBM SPSS Modeler Server installation directory. It configures the environment for the server when the server daemon process is *manually* started. Run it when you want to manually start and stop the server. Edit it when you need to change the configuration for manual startup.
 - auto.sh. This is a script that configures your system to start the server daemon process automatically at boot time. Run it once to configure your system for automatic startup. You do not need to edit it. The script is located in the IBM SPSS Modeler Server installation directory.
 - rc.modeler. When you run auto.sh, this script is created in a location that depends on your server's operating system. It configures the environment for the server when it is *automatically* started. Edit it when you need to change the configuration for automatic startup.
-

Automatically Starting and Stopping IBM SPSS Modeler Server

IBM® SPSS® Modeler Server must be started as a daemon process. The installation program includes a script (*auto.sh*) that you can run to configure your system to automatically stop and restart IBM SPSS Modeler Server.

To configure the system for automatic startup and shutdown

1. Log on as root.
2. Change to the IBM SPSS Modeler Server installation directory.
3. Run the script. At the UNIX prompt, type:

```
./auto.sh
```

An automatic startup script, *rc.modeler*, is created in the location shown in the table above. The operating system will use *rc.modeler* to start the IBM SPSS Modeler Server daemon process whenever the server computer is rebooted. The operating system will also use *rc.modeler* to stop the daemon whenever the system is shut down.

Manually Starting and Stopping IBM SPSS Modeler Server

You can manually start and stop IBM® SPSS® Modeler Server by running the *modelersrv.sh* script.

To manually start and stop IBM SPSS Modeler Server

1. Change to the IBM SPSS Modeler Server installation directory.
2. To start the server, at the UNIX command prompt, type:

```
./modelersrv.sh start
```

3. To stop the server, at the UNIX command prompt type:

```
./modelersrv.sh stop
```

Editing Scripts

If you use both manual and automatic startup, make parallel changes in both *modelersrv.sh* and *rc.modeler*. If you use only manual startup, make changes in *modelersrv.sh*. If you use only automatic startup, make changes in *rc.modeler*.

To edit the scripts

1. Stop IBM® SPSS® Modeler Server. (See the topic [Manually Starting and Stopping IBM SPSS Modeler Server](#) for more information.)
2. Locate the appropriate script. (See the topic [Scripts](#) for more information.)
3. Open the script in a text editor, make changes, and save the file.
4. Start IBM SPSS Modeler Server, either automatically (by restarting the server computer) or manually.

Controlling Permissions on File Creation

IBM® SPSS® Modeler Server creates temporary files with read, write, and execute permissions for everyone. You can override this default by editing the **UMASK** setting in the startup script, either in *modelersrv.sh*, *rc.modeler*, or in both. (For more information, see [Editing Scripts](#) above.) We recommend 077 as the most restrictive **UMASK** setting to use. Settings that are more restrictive could cause permissions problems for IBM SPSS Modeler Server.

IBM SPSS Modeler Server and the data access pack

If you want to use the ODBC drivers with IBM® SPSS® Modeler Server, the ODBC environment must be configured by *odbc.sh* when the IBM SPSS Modeler Server process starts. You do this by editing the appropriate IBM SPSS Modeler start up script, either in *modelersrv.sh*, *rc.modeler*, or in both. (See [Editing Scripts](#) for more information.)

For more information, see the Technical Support web site at <http://www.ibm.com/support>. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

To configure ODBC to start with IBM SPSS Modeler Server

1. Stop the IBM SPSS Modeler Server host if it is running.
2. Download the relevant compressed TAR archive for the platform on which you have IBM SPSS Modeler Server installed. Make sure that you download the correct drivers for your installed version of IBM SPSS Modeler Server. Copy the file to the location where you want to install the ODBC drivers (for example, /usr/spss/odbc).
3. Extract the TAR archive file using **tar -xvof**.
4. Run the *setodbcpath.sh* script that is extracted from the archive.
5. Edit the script *odbc.sh* to add the definition of ODBCINI to the bottom of this script and export it, for example:

```
ODBCINI=/usr/spss/odbc/odbc.ini; export ODBCINI
```

ODBCINI must point to the full pathname of the *odbc.ini* file that you want IBM SPSS Modeler to read to get a list of the ODBC datasources that you define (a default *odbc.ini* is installed with the drivers).

6. Save *odbc.sh*.
7. (64-bit IBM SPSS Modeler Server installations only; for other installations, continue from the next step) Define and export **LD_LIBRARY_PATH_64** in *odbc.sh*:

```

if [ "$LD_LIBRARY_PATH_64" = "" ]; then
    LD_LIBRARY_PATH_64=<library_path>
else
    LD_LIBRARY_PATH_64=<library_path>:$LD_LIBRARY_PATH_64
fi
export LD_LIBRARY_PATH_64

```

where library_path is the same as for the LD_LIBRARY_PATH definition already in the script that has been initialized with your installation path (for example, /usr/spss/odbc/lib). The easiest way to do this is to copy the `if` and `export` statements for LD_LIBRARY_PATH in your odbc.sh file, append them to the end of the file, and then replace the "LD_LIBRARY_PATH" strings in the newly appended `if` and `export` statements with "LD_LIBRARY_PATH_64".

As an example, your final odbc.sh file on a 64-bit IBM SPSS Modeler Server installation might look like this:

```

if [ "$LD_LIBRARY_PATH" = "" ]; then
    LD_LIBRARY_PATH=/usr/spss/odbc/lib
else
    LD_LIBRARY_PATH=/usr/spss/odbc/lib:$LD_LIBRARY_PATH
fi
export LD_LIBRARY_PATH
if [ "$LD_LIBRARY_PATH_64" = "" ]; then
    LD_LIBRARY_PATH_64=/usr/spss/odbc/lib
else
    LD_LIBRARY_PATH_64=/usr/spss/odbc/lib:$LD_LIBRARY_PATH_64
fi
export LD_LIBRARY_PATH_64
ODBCINI=/usr/spss/odbc/odbc.ini; export ODBCINI

```

Remember to export LD_LIBRARY_PATH_64, as well as defining it with the `if` loop.

8. Edit the odbc.ini file that you defined earlier using \$ODBCINI. Define the data source names that you require (these depend on the database that you are accessing).
9. Save the odbc.ini file.
10. Configure IBM SPSS Modeler Server to use these drivers. To do so, edit modelersrv.sh and add the following line immediately below the line that defines SCLEMNAME:

```
. <odbc.sh_path>
```

where odbc.sh_path is the full path to the odbc.sh file that you edited near the beginning of this procedure, for example:

```
. /usr/spss/odbc/odbc.sh
```

Note: The syntax is important here; be sure to leave a space between the first period and the path to the file.

11. Save modelersrv.sh.

Important: For the SDAP driver to work on Db2 on z/OS, you must grant access to SYSIBM.SYSPACKSTMT.

To test the connection

1. Restart IBM SPSS Modeler Server.
2. Connect to IBM SPSS Modeler Server from a client.
3. On the client, add a Database source node to the canvas.
4. Open the node and verify that you can see the data source names that you defined in the odbc.ini file earlier in the configuration procedure.

If you do not see what you expect here, or you get errors when you try to connect to a data source that you have defined, follow the Troubleshooting procedure. See the topic [Troubleshooting ODBC configuration](#) for more information.

To configure ODBC to start with IBM SPSS Modeler Solution Publisher Runtime

When you can successfully connect to the database from IBM SPSS Modeler Server, you can configure an IBM SPSS Modeler Solution Publisher Runtime installation on the same server by referencing the same odbc.sh script from the startup script of IBM SPSS Modeler Solution Publisher Runtime.

1. Edit the modelerrun script in IBM SPSS Modeler Solution Publisher Runtime to add the following line immediately above the last line of the script:

```
. <odbc.sh_path>
```

where odbc.sh_path is the full path to the odbc.sh file that you edited near the beginning of this procedure, for example:

```
. /usr/spss/odbc/odbc.sh
```

Note: The syntax is important here. Be sure to leave a space between the first period and the path to the file.

2. Save the modelerrun script file.

3. By default, the DataDirect Driver Manager is not configured for IBM SPSS Modeler Solution Publisher Runtime to use ODBC on UNIX systems. To configure UNIX to load the DataDirect Driver Manager, enter the following commands (where sp_install_dir is the installation directory of Solution Publisher Runtime):

```
cd sp_install_dir  
rm -f libspssodbc.so  
ln -s libspssodbc_datadirect.so libspssodbc.so
```

To configure ODBC to start with IBM SPSS Modeler Batch

No configuration of the IBM SPSS Modeler Batch script is necessary for ODBC. This is because you connect to IBM SPSS Modeler Server from IBM SPSS Modeler Batch in order to run streams. Ensure that the IBM SPSS Modeler Server ODBC configuration has been made and is working correctly, as described earlier in this section.

To add or edit a data source name

1. Edit the odbc.ini file to include the new or changed name.
2. Test the connection as described earlier in this section.

When the connection with IBM SPSS Modeler Server is working correctly, the new or changed data source should also work correctly with IBM SPSS Modeler Solution Publisher Runtime and IBM SPSS Modeler Batch.

SQL Server support with the Data Access Pack driver

The ODBC configuration for SQL Server must have the **Enable_QuotedIdentifiers** ODBC connection attribute set to **Yes** (the default for this driver is **No**). On UNIX this attribute is configured in the system information file (odbc.ini) using the **QuotedID** option.

- [Troubleshooting ODBC configuration](#)
- [Library paths](#)

Troubleshooting ODBC configuration

No data sources listed, or random text displayed

If you open a Database source node and the list of available data sources is empty or contains unexpected entries, it may be due to a problem with the startup script.

1. Check that \$ODBCINI is defined within *modelersrv.sh*, either explicitly in the script itself, or in the *odbc.sh* script that is referenced in *modelersrv.sh*.
2. In the latter case, ensure that ODBCINI points to the full path to the *odbc.ini* file that you have used to define your ODBC data sources.
3. If the path specification in ODBCINI is correct, check the value of \$ODBCINI that is being used in the IBM® SPSS® Modeler Server environment by echoing the variable from within *modelersrv.sh*. To do so, add the following line to *modelersrv.sh* after the point where you define ODBCINI:

```
echo $ODBCINI
```

4. Save and then execute *modelersrv.sh*. The value of \$ODBCINI that is being set in the IBM SPSS Modeler Server environment is written to *stdout* for verification.
5. If no value at all is returned to *stdout*, and you are defining \$ODBCINI in the *odbc.sh* script that you are referencing from *modelersrv.sh*, check that the referencing syntax is correct. It should be:

```
. <odbc.sh_path>
```

where *odbc.sh_path* is the full path to the *odbc.sh* file that you edited near the beginning of this procedure, for example:

```
. /usr/spss/odbc/odbc.sh
```

Note: The syntax is important here; be sure to leave a space between the first period and the path to the file.

When the correct value is echoed to *stdout* on running *modelersrv.sh*, you should be able to see the data source names in the Database source node when you restart IBM SPSS Modeler Server and connect to it from the client.

IBM SPSS Modeler client hangs on clicking Connect in Database Connections dialog box

This behavior can be caused by your library path not being correctly set to include the path to the ODBC libraries. The library path is defined by \$LD_LIBRARY_PATH (and \$LD_LIBRARY_PATH_64 on 64-bit versions).

To see the value of the library path in the IBM SPSS Modeler Server daemon environment, echo the value of the appropriate environment variable from within *modelersrv.sh*, after the line where you are appending the ODBC library path to the library path, and execute the script. The library path value will be echoed to the terminal when you next execute the script.

If you are referencing *odbc.sh* from *modelersrv.sh* to set up your IBM SPSS Modeler Server ODBC environment, echo your library path value from the line after the one where you reference the *odbc.sh* script. To echo the value, add the following line to the script, then save and execute the script file:

```
echo $<library_path_variable>
```

where *<library_path_variable>* is the appropriate library path variable for your server operating system.

The returned value of your library path must include the path to the *lib* subdirectory of your ODBC installation. If it does not, append this location to the file.

If you are running the 64-bit version of IBM SPSS Modeler Server, *\$LD_LIBRARY_PATH_64* will override *\$LD_LIBRARY_PATH* if it is set. If you are having this problem on one of these 64-bit platforms, echo *LD_LIBRARY_PATH_64* as well as *\$LD_LIBRARY_PATH* from *modelersrv.sh* and, if required, set *\$LD_LIBRARY_PATH_64* so that it includes the *lib* subdirectory of your ODBC installation, and export the definition.

Data source name not found and no default driver specified

If you see this error on clicking Connect in the Database Connections dialog box, it usually indicates that your *odbc.ini* file is incorrectly defined. Check that the data source name (DSN) as defined within the **[ODBC Data Sources]** section at the top of the file matches the string specified between the square brackets further down in *odbc.ini* to define the DSN. If these are different in any way, you will see this error when you try to connect using the DSN from within IBM SPSS Modeler. Following is an example of an *incorrect* specification:

```
[ODBC Data Sources]
Oracle=Oracle Wire Protocol

...
[Oracle Driver]
Driver=/usr/ODBC/lib/XEora22.so
Description=SPSS 5.2 Oracle Wire Protocol
AlternateServers=
...
```

You need to change one of the two strings in bold so that they match exactly. Doing so should resolve the error.

Specified driver could not be loaded

This error also indicates that the *odbc.ini* file is incorrectly defined. One possibility is that the Driver parameter within the driver stanza is incorrectly set, for example:

```
[ODBC Data Sources]
Oracle=Oracle Wire Protocol

...
[Oracle]
Driver=/nosuchpath/ODBC/lib/XEora22.so
Description=SPSS 5.2 Oracle Wire Protocol
AlternateServers=
```

1. Check that the shared object specified by the Driver parameter exists.
2. Correct the path to the shared object if it is incorrect.
3. If the Driver parameter is specified in this format:

```
Driver=ODBCHOME/lib/XEora22.so
```

this indicates that you have not initialized your ODBC-related scripts. Run the *setodbcpath.sh* script that is installed with the drivers. See the topic [IBM SPSS Modeler Server and the data access pack](#) for more information. When you have run this script, you should see that the string "ODBCHOME" has been substituted with the path to your ODBC installation. This should resolve the issue.

Another cause may be a problem with the driver's library. Use the **ivtestlib** tool provided with ODBC to confirm that the driver can't be loaded. For Connect64, use the **ddtestlib** tool. Correct the problem by setting the library path variable in the startup script.

For example, if the Oracle driver cannot be loaded for a 32-bit installation, follow these steps:

1. Use **ivtestlib** to confirm that the driver cannot be loaded. For example, at the UNIX prompt, type:

```
sh
cd ODBCDIR
./odbc.sh
./bin/ivtestlib MFor815
```

where **ODBCDIR** is replaced by the path to your ODBC installation directory.

2. Read the message to see if there is an error. For example, the message: Load of MFor815.so failed: ld.so.1: bin/ivtestlib: fatal: libclntsh.so: open failed: No such file or directory indicates that the Oracle client library, *libclntsh.so*, is missing or that it is not on the library path.
3. Confirm that the library exists. If it doesn't, reinstall the Oracle client. If the library is there, type the following sequence of commands from the UNIX command prompt:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/bigdisk/oracle/product/8.1.6/lib  
export LD_LIBRARY_PATH  
.bin/ivtestlib Mfor815
```

where */bigdisk/oracle/product/8.1.6/lib* is replaced by the path to *libclntsh.so* and **LD_LIBRARY_PATH** is the library path variable for your operating system.

Note that if you are running IBM SPSS Modeler 64-bit on Linux, the library path variable contains the suffix *_64*. Therefore, the first two lines in the previous example would become:

```
LD_LIBRARY_PATH_64=$LD_LIBRARY_PATH_64:/bigdisk/oracle/product/8.1.6/lib  
export LD_LIBRARY_PATH_64
```

4. Read the message to confirm that the driver can now be loaded. For example, the message: Load of MFor815.so successful, qehandle is 0xFF3A1BE4 indicates that the Oracle client library can be loaded.
5. Correct the library path in the IBM SPSS Modeler startup script.
6. Restart the IBM SPSS Modeler Server with the startup script that you edited (*modelersrv.sh* or *rc.modeler*).

Library paths

The name of the library path variable on Linux 64-bit operating system is **LD_LIBRARY_PATH_64**. Use this value to make appropriate substitutions when you're configuring or troubleshooting on your system.

Configuring and Running SPSS® Modeler Server as a Non-Root Process on UNIX

- [Introduction](#)
- [Configuring as non-root without a private password database](#)
- [Configuring as non-root using a private password database](#)
- [Running SPSS Modeler Server as a non-root user](#)
- [Troubleshooting user authentication failures](#)

Introduction

These instructions provide information on running IBM® SPSS® Modeler Server as a non-root process on UNIX systems.

Running as root. The default installation of IBM SPSS Modeler Server assumes that the server daemon process will run as root. Running as root allows IBM SPSS Modeler to authenticate reliably each user login and start each user session on the corresponding UNIX user account. This ensures that users have access only to their own files and directories.

Running as non-root. Running IBM SPSS Modeler Server as a non-root process means having the real and effective user IDs of the server daemon process set to an account of your choice. All user sessions started by SPSS Modeler Server will use the same UNIX account and this means that any file data read or written by SPSS Modeler is shared by all SPSS Modeler users. Access to database data is not affected because users have to authenticate themselves independently to each of the database data sources they use. Without root privilege, IBM SPSS Modeler operates in one of two ways:

- **Without a private password database.** With this method, SPSS Modeler uses the existing UNIX password database, NIS, or LDAP server that is normally used for user authentication on the UNIX system. See the topic [Configuring as non-root without a private password database](#) for more information.
- **With a private password database.** With this method, SPSS Modeler authenticates users against a private password database, distinct from the UNIX password database, NIS, or LDAP server that is normally used for authentication on UNIX. See the topic [Configuring as non-root using a private password database](#) for more information.

Note: On Linux/UNIX systems where both a non-root and SSL configuration are enabled, SSL security will be reduced. Because all user sessions run under the same credential as each other and as the Modeler Server daemon, the SSL certificate data that should be kept secret will instead be exposed to all users. This allows users to easily bypass the normal protections SSL provides to all other users. See [Securing client/server and server-server communications with SSL](#).

Configuring as non-root without a private password database

To configure IBM® SPSS® Modeler Server to run on a non-root account without the need for a private password database, follow these steps:

1. Open the SPSS Modeler Server *options.cfg* file for editing.
2. Set the option **start_process_as_login_user** to **Y**.
3. Save and close the *options.cfg* file.

By default, SPSS Modeler Server tries each authentication method until it finds one that works. However, if desired you can use the **authentication_methods** option in *options.cfg* to configure the server to try only one specific authentication method. Possible values for the option are **pasw_modeler, gss, pam, sspi, unix, or windows**.

Note that running as non-root is likely to require configuration updates. See the topic [Troubleshooting user authentication failures](#) for more information.

CAUTION:

Do not enable the **start_process_as_login_user** setting and then start the IBM SPSS Modeler Server as **root**. Doing so would mean that, for all users that are connected to the server, their server processes would run as **root**; this is a security risk. Note that the server may stop automatically if you attempt this.

Configuring as non-root using a private password database

If you choose to authenticate users by means of a private password database, all user sessions are started on the same non-root user account.

To configure IBM® SPSS® Modeler Server to run on a non-root account in this way, follow these steps:

1. Create a group to contain all of your users. You can name this group whatever you'd like, but for this example, let's call it *modelerusers*.
2. Create the user account on which to run IBM SPSS Modeler Server. This account is for the sole use of the IBM SPSS Modeler Server daemon process. For this example, let's call it *modelerserv*.

When creating the account, note that:

- The primary group should be the *<modelerusers>* group created previously.
 - The home directory can be the IBM SPSS Modeler installation directory or any other convenient default (consider using something other than the installation directory if you need the account to survive upgrades).
3. Next, configure the startup scripts to start IBM SPSS Modeler Server using the newly created account. Locate the appropriate startup script and open it in a text editor. See the topic [Scripts](#) for more information.
 - a. Change the **umask** setting to allow at least group read access on created files:

```
umask 027
```

4. Edit the server options file, *config/options.cfg*, to specify authentication against the private password database by appending the line:

```
authentication_methods, "pasw_modeler"
```

5. Edit the server options file, *config/options.cfg*, to set the option **start_process_as_login_user** to **Y**.
6. Next, you'll need to create a private password database stored in the file *config/passwords.cfg*. The password file defines the user name/password combinations that are allowed to login to IBM SPSS Modeler. *Note:* These are private to IBM SPSS Modeler and have no connection with the user names and passwords used to login to UNIX. You can use the same user names for convenience, but you cannot use the same passwords.

To create the password file, you will need to use the password utility program, *pwutil*, located in the *bin* directory of the IBM SPSS Modeler Server installation. The synopsis of this program is:

```
pwutil [ username [ password ] ]
```

The program takes a user name and plain-text password and writes the user name and encrypted password to the standard output in a format suitable for inclusion in the password file. For example, to define a user *modeler* with the password “data mining” you would type:

```
bin/pwutil modeler "data mining" > config/passwords.cfg
```

Defining a single user name is sufficient in most cases, where all users log in with the same name and password. However additional users can be created by using the **>>** operator to append each to the file, for example:

```
bin/pwutil modeler "data miner2" >> config/passwords.cfg
```

Note: If a single **>** is used, the contents of *passwords.cfg* will be overwritten each time, replacing any users set previously. Remember that all users share the same UNIX user account regardless.

Note: If you add new users to the private passwords database while SPSS Modeler Server is running, you will need to restart SPSS Modeler Server so that it can recognize the newly defined users. Until you do so, logins will fail for any new users added via `pwutl` since the last restart of SPSS Modeler Server.

7. Recursively change the ownership of the IBM SPSS Modeler installation directory and its contents to be user `<modelerserv>` and group `<modelerusers>` where the names referenced are those you created earlier. For example:

```
chown -R -h modelerserv:modelerusers .
```

8. Consider creating subdirectories in the data directory for your IBM SPSS Modeler users so that they have somewhere to store working data without interference. These directories should be group-owned by the `<modelerusers>` group and have group read, write, and search permissions. For example, to create a working directory for user `bob`:

```
mkdir data/bob
chown bob:modelerusers data/bob
chmod ug=rwx,o= data/bob
```

Additionally, you can set the set-group-ID bit on the directory so that any data files copied into the directory will be automatically group-owned by `<modelerusers>`:

```
chmod g+s data/bob
```

Running SPSS Modeler Server as a non-root user

To run SPSS® Modeler Server as a non-root user, follow these steps:

1. Log in using the non-root user account created earlier.
2. If you are running with the configuration file option `start_process_as_login_user` enabled, you can start, stop, and check the status of SPSS Modeler Server. See the topic [To Start, Stop, and Check Status on UNIX](#) for more information.

End users connect to SPSS Modeler Server by logging in from the client software. You must give end users the information that they need to connect, including the IP address or host name of the server machine.

Troubleshooting user authentication failures

Depending on how the operating system is configured to perform authentication, you may experience failures to log on to SPSS® Modeler Server when running in a non-root configuration. For example, this may occur if your operating system is configured (using the `/etc/nsswitch.conf` file or similar) to check the local shadow password file, rather than use NIS or LDAP. This occurs because SPSS Modeler Server requires read access to the files used to perform authentication, including the `/etc/shadow` file or its equivalent, which stores secure user account information. However, the operating system file permissions are generally set so that `/etc/shadow` is accessible only by the root user. Under these circumstances a non-root process cannot read `/etc/shadow` to validate user passwords, resulting in an authentication error.

There are several ways to resolve this issue:

- Ask your system administrator to configure the operating system to use NIS or LDAP for authentication.
- Change the file permissions on the protected files, for example by granting read access to the `/etc/shadow` file so that the local user account used to run SPSS Modeler Server can access the file. While this workaround might be deemed unsuitable in production environments, it could be temporarily applied to a test environment to verify whether the authorization failure is linked to the operating system configuration.
- Specify an access control list (ACL) for the `/etc/shadow` file.
- Run SPSS Modeler Server as root, to enable the server processes to read the `/etc/shadow` file.

CAUTION:

In this case, ensure that the options.cfg file for SPSS Modeler Server contains the option `start_process_as_login_user, N` to avoid the security issue explained earlier.

Configuring and Running SPSS® Modeler Server with a private password file on Windows

- [Introduction](#)
- [Configuring a private password database](#)

Introduction

These instructions provide information on running IBM® SPSS® Modeler Server using a private password file on Windows systems. With this method, IBM SPSS Modeler authenticates users against a private password database, distinct from the system authentication on Windows.

Configuring a private password database

If you choose to authenticate users by using a private password database, all user sessions are started on the same user account.

To configure SPSS® Modeler Server in this way, follow these steps:

1. Create the user account on which to run SPSS Modeler Server. This account is for the sole use of the SPSS Modeler Server daemon process. You must start the daemon process as that user account in the Log On tab of the SPSS Modeler Server 18.4.0 Service. For this example, let's call it *modelerserv*.
2. Edit the server options file (config/options.cfg) to set the option `start_process_as_login_user` to **Y**
3. Edit the server options file (config/options.cfg) to specify authentication against the private password database by appending the line:

```
authentication_methods, "pasw_modeler"
```

4. Next, you need to create a private password database that is stored in the file config/passwords.cfg. The password file defines the user name/password combinations that are allowed to log in to SPSS Modeler. Note that these combinations are private to SPSS Modeler and have no connection with the user names and passwords that are used to log in to Windows. You can use the same user names for convenience, but you cannot use the same passwords.

To create the password file, you need to use the password utility program, pwutil, in the bin directory of the SPSS Modeler Server installation. The synopsis of this program is:

```
pwutil [ username [ password ] ]
```

The program takes a user name and plain-text password and writes the user name and encrypted password to the standard output in a format suitable for inclusion in the password file. For example, to define a user that is called **modeler** with the password **data mining**, you would use a DOS prompt to navigate to the SPSS Modeler Server installation directory and then type:

```
bin\pwutil modeler "data mining" > config\passwords.cfg
```

Note: Make sure that you have only 1 instance of each user in the file; duplicates prevent SPSS Modeler Server from starting. Defining a single user name is sufficient in most cases, where all users log in with the same name and password. However, more users can be created by using the **>>** operator to append each to the file. For example:

```
bin\pwutil modeler "data miner2" >> config\passwords.cfg
```

Note:

If a single **>** is used, the contents of passwords.cfg are overwritten each time, replacing any users set previously. Remember that all users share the same UNIX user account regardless.

If you add new users to the private passwords database while SPSS Modeler Server is running, you will need to restart SPSS Modeler Server so that it can recognize the newly defined users. Until you do so, logins will fail for any new users added via **pwutil** since the last restart of SPSS Modeler Server.

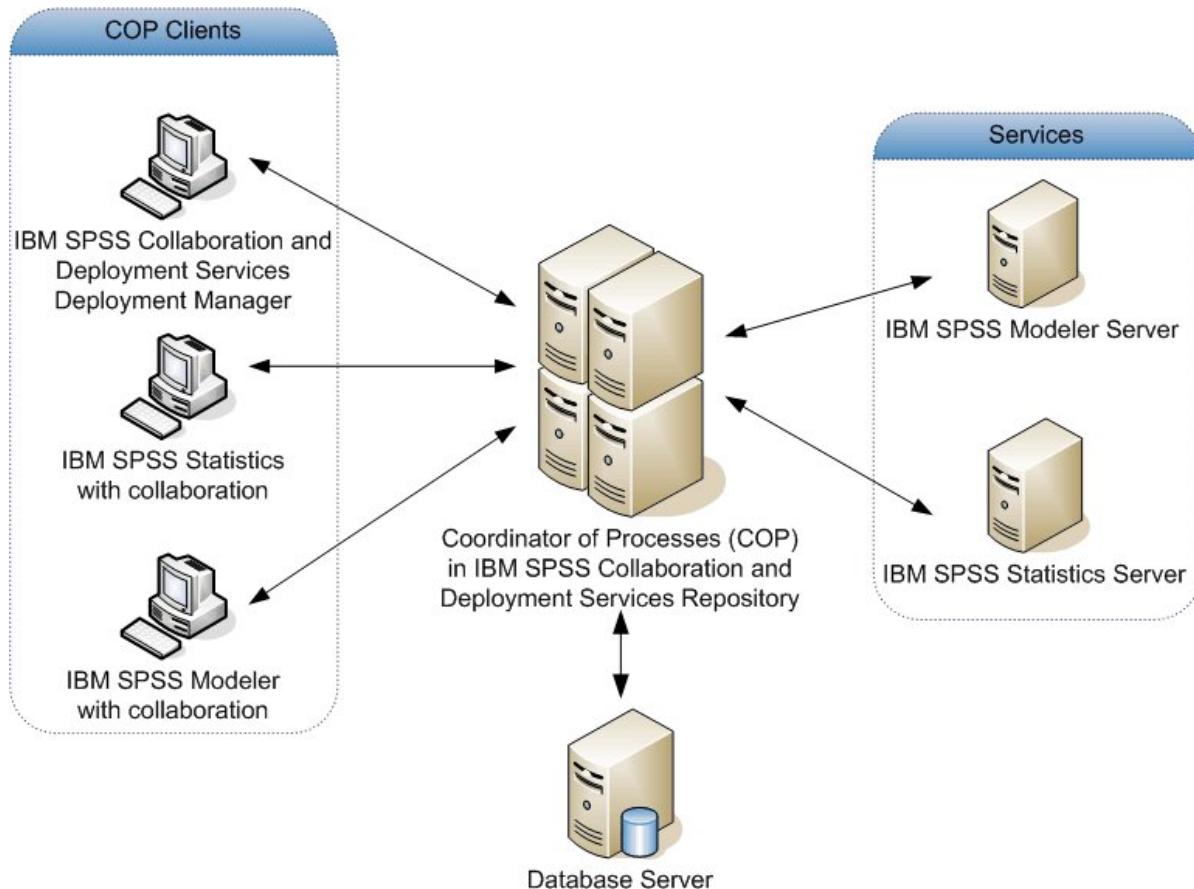
5. Give the user that was created in step 1 full control over the server options file config\options.cfg and the **%ALLUSERSPROFILE%\IBM\SPSS** directory.
6. In the system services, stop the IBM SPSS Modeler Server service and change the Log on from the Local System Account to the user account created in step 1. Then, restart the service.

Load Balancing with Server Clusters

With IBM® SPSS® Collaboration and Deployment Services, a plug-in called the Coordinator of Processes can be used to manage services on the network. The Coordinator of Processes provides server management capabilities designed to optimize client-server communication and processing.

Services to be managed, such as IBM SPSS Statistics Server or IBM SPSS Modeler Server, register with the Coordinator of Processes upon starting and periodically send updated status messages. Services can also store any necessary configuration files in the IBM SPSS Collaboration and Deployment Services Repository and retrieve them when initializing.

Figure 1. Coordinator of Processes Architecture



Executing your IBM SPSS Modeler streams on a server can increase performance. In some cases, you may have only the choice of one or two servers. In other cases, you might be offered a larger choice of servers because there is a substantive difference between each server, such as owner, access rights, server data, test versus production servers, and so on. In addition, if you have the Coordinator of Processes on your network, you might be offered a server cluster.

A server cluster is a group of servers that are interchangeable in terms of configuration and resources. The Coordinator of Processes determines which server is best suited to respond to a processing request, using an algorithm that will balance the load according to several criteria, including the server weights, user priorities, and current processing loads. For more information, see the *Coordinator of Processes Service Developer's Guide* available in the IBM SPSS Collaboration and Deployment Services documentation suite.

Whenever you connect to a server or server cluster in IBM SPSS Modeler, you can enter a server manually or search for a server or cluster using the Coordinator of Processes. See the topic [Connecting to IBM SPSS Modeler Server](#) for more information.

LDAP authentication

These instructions provide basic guidelines on how to configure SPSS® Modeler Server on UNIX to use LDAP authentication, where the identities of the users who will log in to the server are stored in an LDAP directory.

Note: As a prerequisite, the LDAP client software must be correctly configured on the host operating system. For more information, see the original vendor documentation.

Usually no additional configuration is required and the use of LDAP is not apparent to the server. Examples of where no additional changes are required include the following circumstances:

- The LDAP client and server software are configured according to RFC 2307.
- Access to the `passwd` (and, where applicable, `shadow`) database is redirected to LDAP, for example in `nsswitch.conf`.
- Each valid user of SPSS Modeler Server has a `passwd` (and `shadow`) entry that is stored in the LDAP directory.
- The SPSS Modeler Server service is started by using the root account.

There are two sets of circumstances when it might become necessary to configure SPSS Modeler Server specifically for LDAP:

- When the service is started by using an account other than root, the service might not have the authority to authenticate by using the default method. Typically this is because access to the shadow database is restricted.
- When users do not have `passwd` (or `shadow`) entries that are stored in the directory; that is, they do not have user identities that are valid for login to the host system.

The LDAP authentication procedure uses the PAM subsystem and requires that a PAM LDAP module exists and is correctly configured for the host operating system. For more information, see the original vendor documentation.

Complete the following steps to configure SPSS Modeler Server to use LDAP authentication exclusively.

Note: These steps provide the most basic configuration that can be expected to work. More options or alternative settings might be required depending on your operating system and local security policy. For more information, see the original operating documentation.

1. Edit the service configuration file (`options.cfg`) and add (or edit) the line: `authentication_methods, pam`. This line instructs the server to use PAM authentication in preference to the default authentication.
2. Provide a PAM configuration for the SPSS Modeler Server service; which often requires root privileges. The service is identified by the name `modelerserver`.
3. On a Linux type system, which uses `/etc/pam.d`, create a file in that directory with the name `modelerserver` and add content similar to the following example:

```
# IBM SPSS Modeler Server
auth required pam_ldap.so
account required pam_ldap.so
password required pam_deny.so
session required pam_deny.so
```

4. The names of the referenced PAM modules vary by operating system; confirm the modules that are required for your host operating system.

Note: The lines in steps 3 specify that SPSS Modeler Server must refer to the PAM LDAP module for authentication and account management. However, changing of passwords and session management are not supported so these actions are not allowed. If account management is not required or is inappropriate, change the relevant line to permit all requests, as in the following example:

```
# IBM SPSS Modeler Server
auth required pam_ldap.so
account required pam_permit.so
password required pam_deny.so
session required pam_deny.so
```

Overview

- [IBM SPSS Collaboration and Deployment Services](#)

IBM® SPSS® Collaboration and Deployment Services is an enterprise-level application that enables widespread use and deployment of predictive analytics.

- [System architecture](#)

In general, IBM SPSS Collaboration and Deployment Services consists of a single, centralized IBM SPSS Collaboration and Deployment Services Repository that serves a variety of clients, using execution servers to process analytical assets.

- [Working with files](#)

IBM SPSS Collaboration and Deployment Services

IBM® SPSS® Collaboration and Deployment Services is an enterprise-level application that enables widespread use and deployment of predictive analytics.

IBM SPSS Collaboration and Deployment Services provides centralized, secure, and auditable storage of analytical assets and advanced capabilities for management and control of predictive analytic processes, as well as sophisticated mechanisms for delivering the results of analytical processing to users. The benefits of IBM SPSS Collaboration and Deployment Services include:

- Safeguarding the value of analytical assets
- Ensuring compliance with regulatory requirements
- Improving the productivity of analysts
- Minimizing the IT costs of managing analytics

IBM SPSS Collaboration and Deployment Services allows you to securely manage diverse analytical assets and fosters greater collaboration among those developing and using them. Furthermore, the deployment facilities ensure that people get the information they need to take timely, appropriate action.

- [Collaboration](#)

Collaboration refers to the ability to share and reuse analytic assets efficiently, and is the key to developing and implementing analytics across an enterprise.

- [Deployment](#)

To realize the full benefit of predictive analytics, the analytic assets need to provide input for business decisions. Deployment bridges the

gap between analytics and action by delivering results to people and processes on a schedule or in real time.

Collaboration

Collaboration refers to the ability to share and reuse analytic assets efficiently, and is the key to developing and implementing analytics across an enterprise.

Analysts need a location in which to place files that should be made available to other analysts or business users. That location needs a version control implementation for the files to manage the evolution of the analysis. Security is required to control access to and modification of the files. Finally, a backup and restore mechanism is needed to protect the business from losing these crucial assets.

To address these needs, IBM® SPSS® Collaboration and Deployment Services provides a repository for storing assets using a folder hierarchy similar to most file systems. Files stored in the IBM SPSS Collaboration and Deployment Services Repository are available to users throughout the enterprise, provided those users have the appropriate permissions for access. To assist users in finding assets, the repository offers a search facility.

Analysts can work with files in the repository from client applications that leverage the service interface of IBM SPSS Collaboration and Deployment Services. Products such as IBM SPSS Statistics and IBM SPSS Modeler allow direct interaction with files in the repository. An analyst can store a version of a file in development, retrieve that version at a later time, and continue to modify it until it is finalized and ready to be moved into a production process. These files can include custom interfaces that run analytical processes allowing business users to take advantage of an analyst's work.

The use of the repository protects the business by providing a central location for analytical assets that can be easily backed-up and restored. In addition, permissions at the user, file, and version label levels control access to individual assets. Version control and object version labels ensure the correct versions of assets are being used in production processes. Finally, logging features provide the ability to track file and system modifications.

Related information

- [Deployment](#)
-

Deployment

To realize the full benefit of predictive analytics, the analytic assets need to provide input for business decisions. Deployment bridges the gap between analytics and action by delivering results to people and processes on a schedule or in real time.

In IBM® SPSS® Collaboration and Deployment Services, individual files stored in the repository can be included in processing **jobs**. Jobs define an execution sequence for analytical artifacts and can be created with IBM SPSS Deployment Manager. The execution results can be stored in the repository, on a file system, or delivered to specified recipients. Results stored in the repository can be accessed by any user with sufficient permissions using the IBM SPSS Collaboration and Deployment Services Deployment Portal interface. The jobs themselves can be triggered according to a defined schedule or in response to system events.

In addition, the scoring service of IBM SPSS Collaboration and Deployment Services allows analytical results from deployed models to be delivered in real time when interacting with a customer. An analytical model configured for scoring can combine data collected from a current customer interaction with historical data to produce a score that determines the course of the interaction. The service itself can be leveraged by any client application, allowing the creation of custom interfaces for defining the process.

The deployment facilities of IBM SPSS Collaboration and Deployment Services are designed to easily integrate with your enterprise infrastructure. Single sign-on reduces the need to manually provide credentials at various stages of the process. Moreover, the system can be configured to be compliant with Federal Information Processing Standard Publication 140-2.

Note: If a SPSS Modeler stream contains a node that uses a list type, the branch containing that node does not support the scoring service.

Related information

- [Collaboration](#)
-

System architecture

In general, IBM® SPSS® Collaboration and Deployment Services consists of a single, centralized IBM SPSS Collaboration and Deployment Services Repository that serves a variety of clients, using execution servers to process analytical assets.

Figure 1. IBM SPSS Collaboration and Deployment Services Architecture

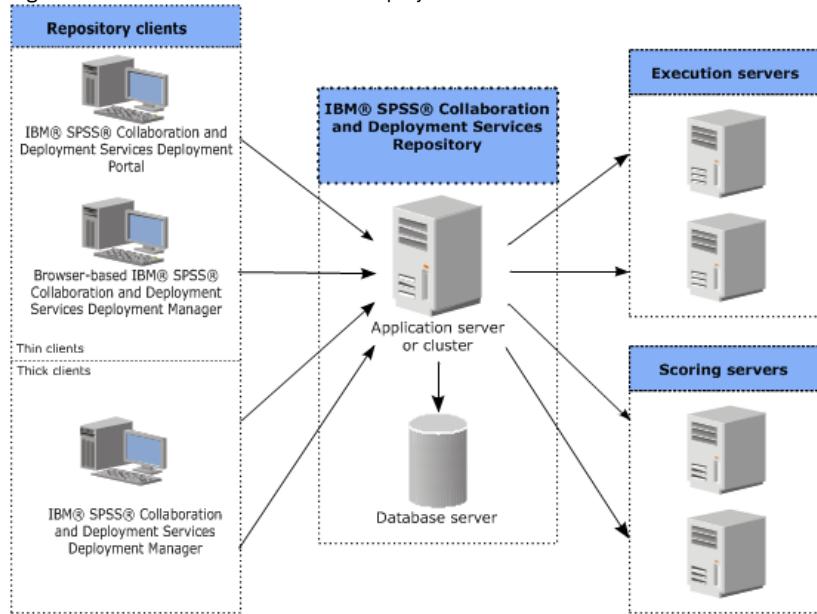


Figure 2. IBM SPSS Collaboration and Deployment Services Architecture

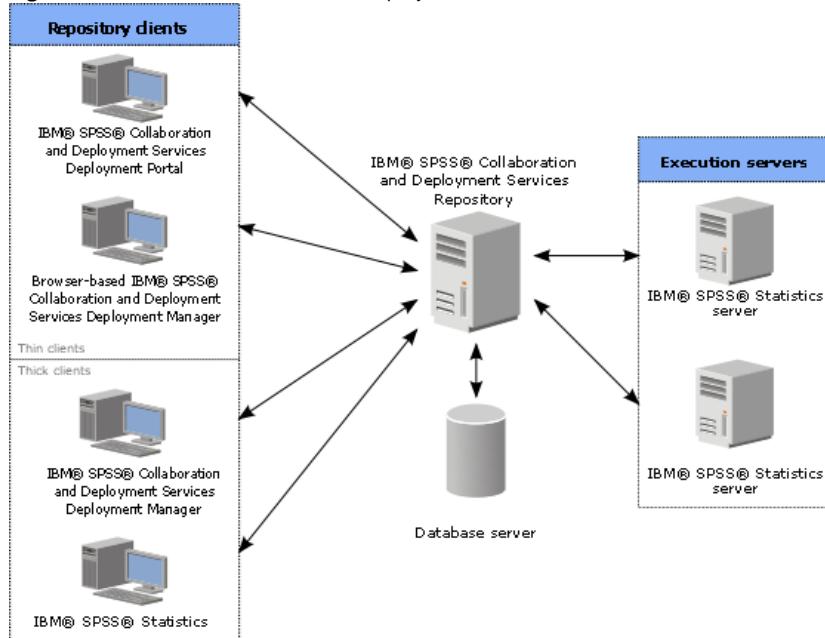
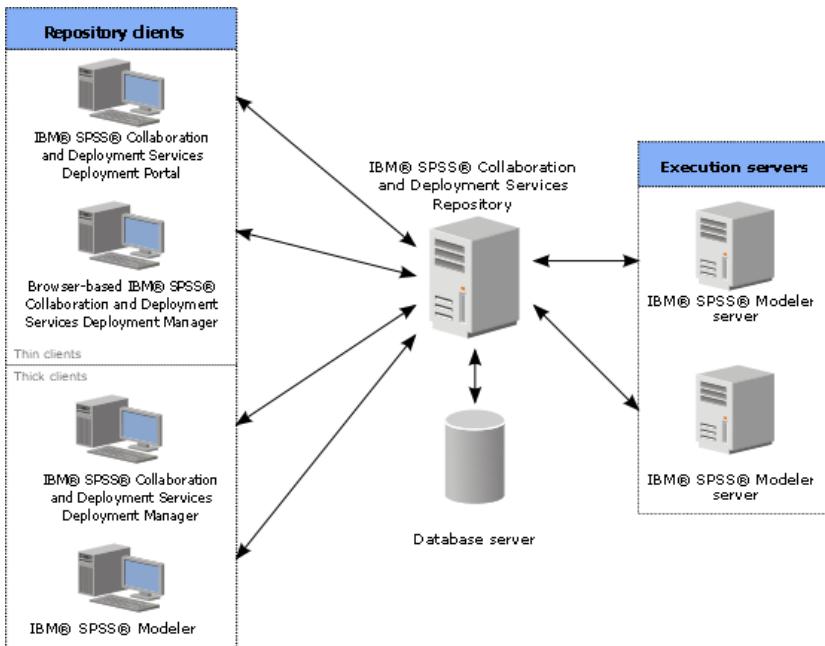


Figure 3. IBM SPSS Collaboration and Deployment Services Architecture



IBM SPSS Collaboration and Deployment Services consists of the following components:

- IBM SPSS Collaboration and Deployment Services Repository for analytical artifacts
- IBM SPSS Statistics
- IBM SPSS Modeler
- IBM SPSS Deployment Manager
- IBM SPSS Collaboration and Deployment Services Deployment Portal
- Browser-based IBM SPSS Deployment Manager

- **IBM SPSS Collaboration and Deployment Services Repository**

The repository provides a centralized location for storing analytical assets, such as models and data. The repository requires an installation of a relational database, such as IBM Db2, Microsoft SQL Server, or Oracle.

- **IBM SPSS Modeler with collaboration**

- **IBM SPSS Deployment Manager**

IBM SPSS Deployment Manager is a client application for IBM SPSS Collaboration and Deployment Services Repository that enables users to schedule, automate, and execute analytical tasks, such as updating models or generating scores.

- **IBM SPSS Collaboration and Deployment Services Deployment Portal**

IBM SPSS Collaboration and Deployment Services Deployment Portal is a thin-client interface for accessing the repository. Unlike the browser-based IBM SPSS Deployment Manager, which is intended for administrators, IBM SPSS Collaboration and Deployment Services Deployment Portal is a web portal serving a variety of users.

- **Browser-based IBM SPSS Deployment Manager**

- **Execution servers**

Execution servers provide the ability to execute resources stored within the repository. When a resource is included in a job for execution, the job step definition includes the specification of the execution server used for processing the step. The execution server type depends on the resource.

Related information

- [IBM SPSS Collaboration and Deployment Services](#)
-

IBM SPSS Collaboration and Deployment Services Repository

The repository provides a centralized location for storing analytical assets, such as models and data. The repository requires an installation of a relational database, such as IBM Db2, Microsoft SQL Server, or Oracle.

The repository includes facilities for:

- Security
- Version control
- Searching
- Auditing

Configuration options for the repository are defined using the IBM® SPSS® Deployment Manager or the browser-based IBM SPSS Deployment Manager. The contents of the repository are managed with the Deployment Manager and accessed with the IBM SPSS Collaboration and Deployment Services Deployment Portal.

IBM SPSS Modeler with collaboration

IBM® SPSS® Modeler with collaboration allows interaction with the IBM SPSS Collaboration and Deployment Services Repository from within the IBM SPSS Modeler interface. Files can be stored and retrieved directly from IBM SPSS Modeler.

In addition, IBM SPSS Modeler streams stored in the repository can be executed as steps within jobs. A job can contain any number of steps, with each step corresponding to a separate file. Relationships defined between the steps determine the processing flow. The job can be scheduled to execute at a specific time, according to a recurrence pattern, or in response to a defined event. Moreover, notifications can be sent to specified recipients to report on individual step and overall job execution status.

Collaboration between IBM SPSS Collaboration and Deployment Services and IBM SPSS Modeler is enabled through the use of adapters. These adapters are installed into the IBM SPSS Collaboration and Deployment Services environment to add the product-specific features. For more information, consult the IBM SPSS Modeler documentation.

Note: If you have a job containing an SPSS Modeler stream that uses an Analytic Server source node, you must allow a direct connection between the IBM SPSS Collaboration and Deployment Services Server and Analytic Server. Otherwise, the job will fail if the firewall blocks the connection between the two servers.

IBM SPSS Deployment Manager

IBM® SPSS® Deployment Manager is a client application for IBM SPSS Collaboration and Deployment Services Repository that enables users to schedule, automate, and execute analytical tasks, such as updating models or generating scores.

The client application allows a user to perform the following tasks:

- View any existing files within the system, including and data files
- Import files into the repository
- Schedule jobs to be executed repeatedly using a specified recurrence pattern, such as quarterly or hourly
- Modify existing job properties
- Determine the status of a job
- Specify email notification of job status

In addition, the client application allows users to perform administrative tasks for IBM SPSS Collaboration and Deployment Services, including:

- Manage users
- Configure security providers
- Assign roles and actions

Browser-based IBM SPSS Deployment Manager

The browser-based IBM SPSS Deployment Manager is a thin-client interface for performing setup and system management tasks, including:

- Setting system configuration options
- Configuring security providers
- Managing MIME types

Non-administrative users can perform any of these tasks provided they have the appropriate actions associated with their login credentials. The actions are assigned by an administrator.

You typically access the browser-based IBM SPSS Deployment Manager at the following URL:

`http://<host IP address>:<port>/security/login`

Note: An IPv6 address must be enclosed in square brackets, such as `[3ffe:2a00:100:7031::1]`.

If your environment is configured to use a custom context path for server connections, include that path in the URL.

`http://<host IP address>:<port>/<context path>/security/login`

IBM SPSS Collaboration and Deployment Services Deployment Portal

IBM® SPSS® Collaboration and Deployment Services Deployment Portal is a thin-client interface for accessing the repository. Unlike the browser-based IBM SPSS Deployment Manager, which is intended for administrators, IBM SPSS Collaboration and Deployment Services Deployment Portal is a web portal serving a variety of users.

The web portal includes the following functionality:

- Browsing the repository content by folder
- Opening published content
- Running jobs
- Generating scores using models stored in the repository
- Searching repository content
- Viewing content properties
- Accessing individual user preferences, such as email address and password, general options, subscriptions, and options for output file formats

You typically access the home page at the following URL:

`http://<host IP address>:<port>/peb`

Note: An IPv6 address must be enclosed in square brackets, such as `[3ffe:2a00:100:7031::1]`.

If your environment is configured to use a custom context path for server connections, include that path in the URL.

`http://<host IP address>:<port>/<context path>/peb`

Browser-based IBM SPSS Deployment Manager

The browser-based IBM® SPSS® Deployment Manager is a thin-client interface for performing setup and system management tasks, including:

- Configuring the system.
- Configuring security providers.
- Managing MIME types.

Non-administrative users can perform any of these tasks provided they have the appropriate actions associated with their login credentials. The actions are assigned by an administrator.

Related information

- [IBM SPSS Collaboration and Deployment Services](#)
- [IBM SPSS Collaboration and Deployment Services Repository](#)
- [IBM SPSS Modeler with collaboration](#)
- [IBM SPSS Deployment Manager](#)
- [IBM SPSS Collaboration and Deployment Services Deployment Portal](#)
- [Execution servers](#)
- [Products with Collaboration](#)
- [Working with files](#)

Execution servers

Execution servers provide the ability to execute resources stored within the repository. When a resource is included in a job for execution, the job step definition includes the specification of the execution server used for processing the step. The execution server type depends on the resource.

Execution servers currently supported by IBM® SPSS® Collaboration and Deployment Services include:

- **SAS**. The SAS execution server is the SAS executable file `sas.exe`, included with Base SAS® Software. Use this execution server to process SAS syntax files.
- **Remote Process**. A remote process execution server allows processes to be initiated and monitored on remote servers. When the process completes, it returns a success or failure message. Any machine acting as a remote process server must have the necessary infrastructure installed for communicating with the repository.

Note: The IBM SPSS Collaboration and Deployment Services Remote Process Server has a default thread pool core size of 16, which allows a maximum of 16 concurrent jobs to be executed on a single remote process server. Any concurrent jobs in excess of 16 must wait in the queue until the available thread pool has free resources. To manually configure the IBM SPSS Collaboration and Deployment Services Remote Process Server thread pool core size, add the following JVM option (with a user defined value) to the remote process server's startup script: `prms.thread.pool.coreSize=<user defined value>`

For more information regarding the start-up script, see the "Starting and stopping the remote process server" section in the IBM SPSS Collaboration and Deployment Services Remote Process Server guide.

Execution servers that process other specific types of resources can be added to the system by installing the appropriate adapters. For information, consult the documentation for those resource types.

The IBM SPSS Modeler execution server is IBM SPSS Modeler Server, which permits distributed analysis for data mining and model building. This execution server requires the specification of user credentials under which the processing occurs.

Two types of execution servers are available for processing IBM SPSS Statistics syntax files. The first is a remote execution server that corresponds to the IBM SPSS Statistics server product. This execution server requires the specification of user credentials under which the syntax processing occurs. The other type of execution server is local to the repository and corresponds to the batch facility included with the IBM SPSS Statistics server product. However, unlike the IBM SPSS Statistics server product, the batch facility can be run from the command line and requires no user credentials for execution.

To allow load balancing, two or more execution servers can be grouped together in a server cluster. When a job step uses a cluster for execution, IBM SPSS Collaboration and Deployment Services determines which managed server in the cluster is best suited to handle processing requests at that time. For more information, consult the IBM SPSS Deployment Manager documentation.

During job creation, assign an execution server or server cluster to each step included in the job. When the job executes, the repository uses the specified execution servers to perform the corresponding analyses.

Working with files

In IBM® SPSS® Deployment Manager, the general process for working with files involves:

1. Define an execution server for processing the if an appropriate definition does not already exist.
 2. Add the to a job as a job step.
 3. Specify the job properties, including any schedules for execution.
-

Server definitions

Executing an IBM® SPSS® Collaboration and Deployment Services Repository resource as a job step requires the specification of an appropriate corresponding server to process the instructions contained in the job step. The connection information for such a server is specified within a server definition.

Server definitions can be classified as either execution servers or repository servers.

- Execution servers process the contents of an IBM SPSS Collaboration and Deployment Services Repository resource. The execution server type must correspond to the resource type being processed. A SAS An IBM SPSS Modeler An IBM SPSS Statistics job step, for example, requires a SAS an IBM SPSS Modeler an IBM SPSS Statistics server definition.
- A repository server corresponds to an IBM SPSS Collaboration and Deployment Services repository installation. A server of this type is typically used by job steps that need to return result artifacts to a repository.

Server definitions are contained in the *Resource Definitions* folder of the Content Explorer. Specifically, they are defined in the *Servers* subfolder.

- [Adding new server definitions](#)
When you create a server definition, specify the name, type, location, and parameters for the server.
 - [Modifying server definitions](#)
To modify a server definition, double-click the server in the Content Explorer.
-

Adding new server definitions

To add a new server:

1. In the Content Explorer, open the *Resource Definitions* folder.
2. Click the *Servers* folder.
3. From the File menu, choose:

New > Server Definition

The Add New Server Definition wizard opens. Alternatively, the new server definition dialog box can be accessed by clicking New next to a server field on the General tab for some steps. The process for defining new servers consists of:

1. Naming the server definition and specifying its type. Note that the server types available depend on which product adapters are installed to the repository.
2. Selecting a location in the *Servers* folder for the definition.
3. Specifying parameters for the server that define connection or execution information. The parameter set depends on the server type.

- **[IBM SPSS Modeler server parameters](#)**

An IBM SPSS Modeler Server Definition specifies the connection parameters for IBM SPSS Modeler servers used to process job steps. IBM SPSS Modeler streams are executed on the IBM SPSS Modeler server.

IBM SPSS Modeler server parameters

An IBM® SPSS® Modeler Server Definition specifies the connection parameters for IBM SPSS Modeler servers used to process job steps. IBM SPSS Modeler streams are executed on the IBM SPSS Modeler server.

1. In the Host field, enter the hostname where the server resides. For example, if you are creating an IBM SPSS Modeler server definition, the host would be the machine that contains your IBM SPSS Modeler server.
2. In the Port field, enter the port number to be used to connect to the host.
3. In the Default Data Path field, enter the path on which you want to place data files.
4. If Secure Socket Layer (SSL) is to be used for the server connection, select This is a secure port.
5. Click Finish. The new definition appears in the *Servers* folder.

Note: When you run a IBM SPSS Collaboration and Deployment Services job in an Evaluation stream, a temporary file is created. By default, the file is saved to the IBM SPSS Modeler Server installation directory. You can change the default data folder that the temp files are saved to when you create the IBM SPSS Modeler Server connection in IBM SPSS Modeler.

Modifying server definitions

To modify a server definition:

1. In the Content Explorer, open the *Resource Definitions* folder.
2. Open the *Servers* folder.
3. Double-click the server to be modified. The Edit Server Definition dialog opens.
4. Modify the server definition parameters as necessary.
5. Click Finish to save your changes.

IBM® SPSS® Modeler Job Steps

- [Working with IBM SPSS Modeler streams](#)
- [Viewing IBM SPSS Modeler job properties](#)
- [Viewing Streams in IBM SPSS Modeler](#)
- [IBM SPSS Modeler Completion Codes](#)
- [IBM SPSS Modeler Stream Limitations](#)
- [Node Types](#)

Working with IBM SPSS Modeler streams

This section describes IBM® SPSS® Modeler streams within the context of IBM SPSS Deployment Manager. IBM SPSS Modeler streams are brought into Deployment Manager fully formed. For more detailed information on how to create and work with streams, refer to the IBM SPSS Modeler documentation.

You can work with IBM SPSS Modeler streams in Deployment Manager. Like any other step, an IBM SPSS Modeler stream must be added to a job before you can execute it using Deployment Manager. Specifically, you can perform the following tasks:

- Importing streams
- Modifying stream parameters
- Executing streams

Note: If you have a job containing an SPSS Modeler stream that uses an Analytic Server source node, you must allow a direct connection between the IBM SPSS Collaboration and Deployment Services Server and Analytic Server. Otherwise, the job will fail if the firewall blocks the connection

between the two servers.

- [IBM SPSS Modeler Server configuration](#)
-

IBM SPSS Modeler Server configuration

Before you begin working with IBM® SPSS® Modeler streams in the Deployment Manager, you need to perform the following configuration tasks:

- Create an IBM SPSS Modeler server definition. See the topic [IBM SPSS Modeler server parameters](#) for more information.
- Define server credentials.

Related information

- [Viewing IBM SPSS Modeler job properties](#)
 - [IBM SPSS Modeler Job Properties - General](#)
 - [IBM SPSS Modeler Job Properties - Data Files](#)
 - [Specifying the input file location](#)
 - [IBM SPSS Modeler Job Properties - ODBC data sources](#)
 - [IBM SPSS Modeler Job Properties - Parameters](#)
 - [IBM SPSS Modeler Job Properties - Results](#)
 - [Viewing output results](#)
 - [Viewing Streams in IBM SPSS Modeler](#)
 - [IBM SPSS Modeler Completion Codes](#)
 - [IBM SPSS Modeler Stream Limitations](#)
-

Viewing IBM SPSS Modeler job properties

When you click an IBM® SPSS® Modeler stream within a job, the following job properties appear:

- General
 - Data files
 - ODBC data sources
 - Parameters
 - Results
 - Cognos import
 - Cognos export
 - Notifications
 - [IBM SPSS Modeler Job Properties - General](#)
 - [IBM SPSS Modeler Job Properties - Data Files](#)
 - [IBM SPSS Modeler Job Properties - Data View](#)
 - [IBM SPSS Modeler Job Properties - ODBC data sources](#)
 - [IBM SPSS Modeler Job Properties - Geo Spatial](#)
 - [IBM SPSS Modeler Job Properties - Parameters](#)
 - [IBM SPSS Modeler Job Properties - Results](#)
 - [IBM SPSS Modeler Job properties - Cognos Import](#)
 - [IBM SPSS Modeler Job properties - Cognos Export](#)
 - [IBM SPSS Modeler Job Properties - Legacy TM1 Import](#)
 - [IBM SPSS Modeler Job Properties - Legacy TM1 Export](#)
 - [IBM SPSS Modeler Job Properties - TM1 Import](#)
 - [IBM SPSS Modeler Job Properties - TM1 Export](#)
 - [IBM SPSS Modeler Job Properties - Analytic Server Import](#)
 - [IBM SPSS Modeler Job Properties - Analytic Server Export](#)
 - [IBM SPSS Modeler Job Properties - Notifications](#)
-

IBM SPSS Modeler Job Properties - General

By default, the General properties tab appears when you click an IBM® SPSS® Modeler stream in the job canvas.

The General properties tab contains the following information:

Job step name. The name of the job step. Typically, the name of the job step is the name of the IBM SPSS Modeler stream, appended with the suffix **_step**. However, you can modify the job step name.

To modify the name of the job step, type the revised name in the Job step name field. Your changes are reflected in the job canvas.

IBM SPSS Modeler Stream. The name and path of the original IBM SPSS Modeler stream. The source stream is not modifiable.

Override Type Use this option to specify whether the settings in either a job step or the node in a stream take priority during execution.

- Job overrides stream If you select this option, when you execute the job, the node properties use the values that you set in the job step.
- Stream overrides job Selecting this option disables the node property controls in the job step; the only exceptions are some of the credential controls. If you change the node properties in the stream, when you execute the job step, the changed node properties are used.

The Override Type options affect the credential settings in the following ways:

- If you choose Job overrides stream, you can edit the credential selection controls, and your selected credentials take effect at run-time.
- If you choose Stream overrides job, there are two possible cases:
 - When you create nodes in SPSS Modeler Client that use credentials, and if the authentication mode is Stored credential, and the credential name exists in IBM SPSS Collaboration and Deployment Services, the credential selection control is unavailable. However, if the credential name does not exist in IBM SPSS Collaboration and Deployment Services, you can use the credential selection control; your selected credentials take effect at run-time.
 - When you create nodes in SPSS Modeler Client that use credentials, if the authentication mode is Username and password, you can use the credential selection control; your selected credentials take effect at run-time.

Object version. The labeled version of the IBM SPSS Modeler stream to use. From the Object version drop-down list, select the labeled version that you want to use.

Iterative Variable List. If the step acts as an iterative consumer, identify the variable from the iterative producer providing the values for the step. An iterative producer step must appear immediately before the step in the job for the step to act as an iterative consumer.

IBM SPSS Modeler server. The IBM SPSS Modeler server or server cluster on which the stream will be executed. The list contains all servers and server clusters currently configured to execute IBM SPSS Modeler steps. To change the server, select from the IBM SPSS Modeler Server drop-down list. To create a new server definition, click New to launch the server definition wizard.

IBM SPSS Modeler login. The credential information used to access the IBM SPSS Modeler server or server cluster. To change the credentials, select a credential definition from the IBM SPSS Modeler Login drop-down list. To define new credentials, click New to launch the credentials definition wizard.

Content Repository Server. The content repository server allows a job to save files to an IBM SPSS Collaboration and Deployment Services Repository. Typically, the content repository server is specified when refreshing models using IBM SPSS Modeler. To specify a content repository server, select a server from the Content Repository Server drop-down list. To create a new server definition, click New to launch the server definition wizard. To generate a content repository server definition based on the current server information, click Generate. A server definition is created and automatically populated in the *Content Repository Server* field.

Content Repository Login. The login information for the content repository server. To specify a content repository login, select a credential from the Content Repository Login drop-down list. To create a new login, click New to launch the content repository login wizard. If single sign-on is not used to connect to the IBM SPSS Collaboration and Deployment Services Repository, click Generate to generate a content repository server login based on existing security settings. A content repository login is created and automatically populated in the *Content Repository Login* field. Login generation is not available when using single sign-on.

Warning expression. Define warnings for job steps connected by a Conditional connector. The warning expression (e.g., `completion_code`, `warning`, or `success`) must be lowercase.

To use warning expressions:

1. Connect two job steps with a Conditional connector. In the Expression field for the conditional connector, type `warning==true`.
2. Navigate to the General tab of the parent job step.
3. In the Warning expression field, specify a warning code--for example, `completion_code==18`. This expression overrides the default warning code, if any.

When the job is run, the system will execute the parent job step. Then the system will evaluate the condition for `warning==true`. If true, the system will look at the warning expression specified and determine if the condition was met. If the condition specified in the warning expression was met, the system proceeds to the next job step.

Type. The way you want to run the stream.

- Run Stream. With this option, you can run the complete stream or, if the stream contains branches, choose one or more branches to run. In the latter case, the Run Options list is displayed. All branches are checked by default, meaning that all branches will execute when the stream runs. Uncheck any branches that you do not want to execute; doing so can improve performance.
Note: If the stream contains a script and is configured to "Run Script" on execution, no execution branches will be displayed in the list in the IBM SPSS Collaboration and Deployment Services job editor.

- Model Management. Choose this option if you want the stream to make use of model management features, such as evaluation, refresh and score. Choose the feature from the Types list that is displayed. **Evaluation Options**

Performance. Specify the threshold percentages for the following categories: Bad or Good. A percentage for Better is not user-defined. The system subtracts the Good percentage from 100 to arrive at the Better threshold. Threshold values are mandatory for evaluation streams.

Metrics. The measurement criteria by which the stream's effectiveness is assessed. Valid values include Accreditation, Accuracy, and Gains. If Gains is selected, then a percentile needs to be specified as well. If Accreditation is selected, then the accreditation step needs to be specified. Valid values include "Collect Statistics" and "Run Evaluation".

Refresh Options

Lists all model-nugget relationships in the stream that can be refreshed using the node names defined in the stream. Select a specific relationship to be refreshed by checking the box before the relationship. Uncheck a relationship to prevent it from being refreshed when the step executes. To select all relationships for refresh, click the Check All button. To unselect all relationships, click the Uncheck All button.

Score Options

Choose the branch or branches you want to designate as scoring branches.

Guidelines for Providing IBM SPSS Modeler and Content Repository Information

For all IBM SPSS Modeler job steps, both IBM SPSS Modeler server and IBM SPSS Modeler login information are required.

Related information

- [Working with IBM SPSS Modeler streams](#)
- [IBM SPSS Modeler Server configuration](#)
- [IBM SPSS Modeler Job Properties - Data Files](#)
- [Specifying the input file location](#)
- [IBM SPSS Modeler Job Properties - ODBC data sources](#)
- [IBM SPSS Modeler Job Properties - Parameters](#)
- [IBM SPSS Modeler Job Properties - Results](#)
- [Viewing output results](#)
- [Viewing Streams in IBM SPSS Modeler](#)
- [IBM SPSS Modeler Completion Codes](#)
- [IBM SPSS Modeler Stream Limitations](#)

IBM® SPSS® Modeler Job Properties - Data Files

The Data Files table contains the following information:

Node Name. The name of the input node that contains the data used by the stream. The name is prefixed by the names of any supernodes containing the node separated by slashes. For example, if the node *MyNode* is within a supernode named *Supernode1*, the name appears as */Supernode1/MyNode*. The node name is not modifiable.

Node Type. The node type as defined in the stream. The node type is not modifiable.

File Name. The name of the input data file. To change the name, click the File Name cell and change the name.

Format. The format of the output file--for example, a comma delimited file. To modify the file format type, click the Format cell. A drop-down arrow appears. Select the format type.

Location. The location of the input data files. To modify location, click in the column and then click the resulting ellipsis button. The Input File Location dialog box opens. Change the location as necessary.

Nodes within locked supernodes are not accessible. They cannot be viewed or modified.

- [Specifying the input file location](#)

For some job steps, you can specify the location of input files used by the step.

Related information

- [Working with IBM SPSS Modeler streams](#)
- [IBM SPSS Modeler Server configuration](#)
- [IBM SPSS Modeler Job Properties - General](#)
- [IBM SPSS Modeler Job Properties - ODBC data sources](#)

- [IBM SPSS Modeler Job Properties - Parameters](#)
 - [IBM SPSS Modeler Job Properties - Results](#)
 - [Viewing output results](#)
 - [Viewing Streams in IBM SPSS Modeler](#)
 - [IBM SPSS Modeler Completion Codes](#)
 - [IBM SPSS Modeler Stream Limitations](#)
-

Specifying the input file location

For some job steps, you can specify the location of input files used by the step.

To define the location of an input file:

1. Type the new path for the data file in the Folder field.
 2. Click OK.
-

IBM® SPSS® Modeler Job Properties - Data View

If the stream contains a Geo Spatial node, the connection details are displayed here.

Node Name. The name of the Data View node.

Analytic Data View. The analytic data view used.

Label. The label used.

Table Name. The name of the database table used.

Data Access Plan. Select a data access plan from the analytic data view. A data access plan associates the data model tables in an analytic data view with physical data sources. An analytic data view typically contains multiple data access plans. When you change the data access plan in use, you change the data used by your stream. For example, if the analytic data view contains a data access plan for training a model and a data access plan for testing a model, you can switch from training data to testing data by changing the data access plan in use.

- [Data Access Plan](#)
-

IBM® SPSS® Modeler Job Properties - ODBC data sources

The *ODBC Data Sources* table contains the following information.

Node Name. The name of the input node that contains the data used by the stream. The name is prefixed by the names of any supernodes containing the node separated by slashes. For example, if the node *MyNode* is within a supernode named *Supernode1*, the name appears as */Supernode1/MyNode*.

Node Type. The node type as defined in the stream.

ODBC Data Sources. The current ODBC data source name (DSN). To change to a different ODBC data source, click the cell containing current data source name, then click the "..." button that is displayed. Doing so displays a dialog box from where you can choose an existing DSN or create a new one. Note that the job step settings always override the stream; therefore, if you modify the stream to use a different data source you must also edit the job to use the same source, otherwise the job may fail to run.

Credentials. To change the database username and password when changing the ODBC data source, click the cell containing the current credentials, then click the "..." button that is displayed. Doing so displays a dialog box from where you can choose an existing credential definition or create a new one.

Database Table. The database table that corresponds to the node.

Nodes within locked supernodes are not accessible. They cannot be viewed or modified.

- [Changing the ODBC connection](#)
- [Browsing for an ODBC connection](#)
- [Changing the database credentials](#)
- [Browsing for a credential definition](#)

Related information

- [Working with IBM SPSS Modeler streams](#)
 - [IBM SPSS Modeler Server configuration](#)
 - [IBM SPSS Modeler Job Properties - General](#)
 - [IBM SPSS Modeler Job Properties - Data Files](#)
 - [Specifying the input file location](#)
 - [IBM SPSS Modeler Job Properties - Parameters](#)
 - [IBM SPSS Modeler Job Properties - Results](#)
 - [Viewing output results](#)
 - [Viewing Streams in IBM SPSS Modeler](#)
 - [IBM SPSS Modeler Completion Codes](#)
 - [IBM SPSS Modeler Stream Limitations](#)
-

Changing the ODBC connection

You can change the ODBC connection either to one that is already defined, or to one that you create here.

Use Existing ODBC DSN. Click Browse to display a list of existing ODBC DSNs from which you can choose.

Create New ODBC DSN. Select this option and click New to define a new ODBC connection for use with stream job steps.

Browsing for an ODBC connection

This is a list of all the ODBC connections that have been defined on this host and to which you have access.

Choose one from the list and click OK.

Changing the database credentials

When changing the connection, you can switch to use the corresponding database username and password here. You can use an existing credential definition from the repository, or you can create a new definition.

Generate Repository credentials from existing Username and Password. (displayed only if importing a job from Release 4.2.1 of IBM® SPSS® Collaboration and Deployment Services) Click the Generate button to generate repository credentials from an existing username and password if these have been set in the imported job. For example, if the username **sa** already exists, repeatedly clicking Generate will create the usernames **sa1**, **sa2**, and so on.

Use Existing Credential. Click the Browse button to display a list of existing credential definitions from which you can choose.

Create New Credential. Select this option and click New to create a new credential definition for use with the current connection.

Browsing for a credential definition

This is a list of all the credential definitions that have been created on this host and to which you have access.

Choose one from the list and click OK.

IBM® SPSS® Modeler Job Properties - Geo Spatial

If the stream contains a Geo Spatial node, the connection details are displayed here.

Source Type. The datasource type.

File Name. If using a Shape file, specify the file name.

Map Service URL. If using a Map service, specify the URL to the service.

Map ID. Specify the map ID.

IBM SPSS Modeler Job Properties - Parameters

You can modify parameters for IBM® SPSS® Modeler streams using the IBM SPSS Deployment Manager. The Parameters table contains the following information:

Name. The parameter name.

Storage. Describes how the parameter is stored--for example, as a string.

Value. The value of the parameter.

Type. The parameter type.

Parameters within locked supernodes are not accessible. They cannot be viewed or modified.

Related information

- [Working with IBM SPSS Modeler streams](#)
- [IBM SPSS Modeler Server configuration](#)
- [IBM SPSS Modeler Job Properties - General](#)
- [IBM SPSS Modeler Job Properties - Data Files](#)
- [Specifying the input file location](#)
- [IBM SPSS Modeler Job Properties - ODBC data sources](#)
- [IBM SPSS Modeler Job Properties - Results](#)
- [Viewing output results](#)
- [Viewing Streams in IBM SPSS Modeler](#)
- [IBM SPSS Modeler Completion Codes](#)
- [IBM SPSS Modeler Stream Limitations](#)

IBM SPSS Modeler Job Properties - Results

To display output settings for an IBM® SPSS® Modeler job step, click the Results tab.

Node Name. The name of the node that contains the output of the stream processing. The name is prefixed by the names of any supernodes containing the node separated by slashes. For example, if the node *MyNode* is within a supernode named *Supernode1*, the name appears as */Supernode1/MyNode*. The node name is not modifiable.

Node Type. The node type as defined in the stream. The node type is not modifiable.

File Name. The name of the corresponding file. To modify the name, click in the File Name column and type the new name.

Note: The file extension is determined by the selected file format, which is automatically appended to the file name.

Format. The format of the output file. To modify the file format type, click in the Format column. A drop-down arrow appears. Select the format type.

Location. The location of the file. To modify location, open the Results Location dialog box by clicking in the column and then clicking the resulting ellipsis button.

Permissions. Access permissions for the file if saved to the repository. To modify permissions, open the Output Permissions dialog box by clicking in the Permissions column and then clicking the resulting ellipsis button.

Properties. The properties (metadata) of the file. To define properties, open the Output Properties dialog box by clicking in the Properties column and then clicking the resulting ellipsis button.

Nodes within locked supernodes are not accessible. They cannot be viewed or modified.

- [Viewing output results](#)

Related information

- [Working with IBM SPSS Modeler streams](#)
- [IBM SPSS Modeler Server configuration](#)
- [IBM SPSS Modeler Job Properties - General](#)

- [IBM SPSS Modeler Job Properties - Data Files](#)
 - [Specifying the input file location](#)
 - [IBM SPSS Modeler Job Properties - ODBC data sources](#)
 - [IBM SPSS Modeler Job Properties - Parameters](#)
 - [Viewing Streams in IBM SPSS Modeler](#)
 - [IBM SPSS Modeler Completion Codes](#)
 - [IBM SPSS Modeler Stream Limitations](#)
-

Viewing output results

To view the output results of your stream, double-click on the results that you want to view in the Results column of the Job Step History table. The results are opened in a separate window.

Related information

- [Working with IBM SPSS Modeler streams](#)
 - [IBM SPSS Modeler Server configuration](#)
 - [Viewing IBM SPSS Modeler job properties](#)
 - [IBM SPSS Modeler Job Properties - General](#)
 - [IBM SPSS Modeler Job Properties - Data Files](#)
 - [Specifying the input file location](#)
 - [IBM SPSS Modeler Job Properties - ODBC data sources](#)
 - [IBM SPSS Modeler Job Properties - Parameters](#)
 - [Viewing Streams in IBM SPSS Modeler](#)
 - [IBM SPSS Modeler Completion Codes](#)
 - [IBM SPSS Modeler Stream Limitations](#)
-

IBM® SPSS® Modeler Job properties - Cognos Import

If the stream contains an IBM Cognos source node, the Cognos connection details are displayed here.

Node Name. The name of the Cognos source node.

Connection URL. The URL of the Cognos server to which the connection is made.

Package Name. The name of the Cognos package from which the metadata is imported.

Anonymous. Contains Anonymous if anonymous login is used for the Cognos server connection, or Credential if a specific Cognos username and password are used.

Credentials. The username and password (if required) on the Cognos server.

Note: Cognos credentials must be created in a domain, which represents the Cognos namespace ID.

IBM® SPSS® Modeler Job properties - Cognos Export

If the stream contains an IBM Cognos Export node, the Cognos and ODBC connection details are displayed here.

Node Name. The name of the Cognos export node.

Connection URL. The URL of the Cognos server to which the connection is made.

Package Name. The name of the Cognos package used to export the metadata.

Datasource. The name of the Cognos database used to export the data.

Folder. The path and name of the folder on the Cognos server where the export package is created.

Anonymous. Contains Anonymous if anonymous login is used for the Cognos server connection, or Credential if a specific Cognos username and password are used.

Credentials. The username and password (if required) on the Cognos server.

DSN. The database source name (DSN) of the Cognos database.

Table Name. The name of the database table used for the export.

Credentials. The username and password used for connection to the database used for the export.

Note: Cognos credentials must be created in a domain, which represents the Cognos namespace ID.

IBM® SPSS® Modeler Job Properties - Legacy TM1 Import

If the stream contains a Legacy TM1 Import node, the connection details are displayed here.

Node Name. The name of the Legacy TM1 Import node.

TM1 Server. The Cognos TM1 Server name.

Cube. The TM1 cube from which the data will be imported.

View. The view to import from the TM1 cube.

Credential. Credential for the Cognos TM1 Server.

IBM® SPSS® Modeler Job Properties - Legacy TM1 Export

If the stream contains a Legacy TM1 Export node, the connection details are displayed here.

Node Name. The name of the Legacy TM1 Export node.

TM1 Server. The Cognos TM1 Server name.

Cube. The TM1 cube into which the data will be exported.

Credential. Credential for the Cognos TM1 Server.

IBM® SPSS® Modeler Job Properties - TM1 Import

If the stream contains a TM1 Import node, the connection details are displayed here.

Node Name. The name of the TM1 Import node.

Admin Host. The Cognos TM1 administration host.

TM1 Server. The Cognos TM1 Server name.

Cube. The TM1 cube from which the data will be imported.

View. The view to import from the TM1 cube. You can import a public or a private view.

Rows. The rows to import from the TM1 cube.

Columns. The columns to import from the TM1 cube.

Credential. Credential for the Cognos TM1 Server.

IBM® SPSS® Modeler Job Properties - TM1 Export

If the stream contains a TM1 Export node, the connection details are displayed here.

Node Name. The name of the TM1 Export node.

Admin Host. The Cognos TM1 administration host.

TM1 Server. The Cognos TM1 Server name.

Cube. The TM1 cube into which the data will be exported.

Measure. The measure that will be exported.

Credential. Credential for the Cognos TM1 Server.

IBM® SPSS Modeler Job Properties - Analytic Server Import

If the stream contains an Analytic Server Source node, the connection details are displayed here.

Use Default Analytic Server. True or False specifies whether the default Analytic Server connection defined by an administrator in options.cfg is used (True), or if a different Analytic Server is used – the one specified in the stream/job (False).

URL. The URL for the Analytic Server in the format `https://hostname:port/contextroot`, where `hostname` is the IP address or host name of the Analytic Server, `port` is its port number, and `contextroot` is the context root of the Analytic Server.

Tenant. The name of the tenant that the SPSS® Modeler Server is a member of.

Credential. The credential for logging on to Analytic Server.

Service Principal Name. The Kerberos service principal name.

Config File Path. The Kerberos service config file path.

Data Source. The data source name on Analytic Server.

IBM® SPSS Modeler Job Properties - Analytic Server Export

If the stream contains an Analytic Server Export node, the connection details are displayed here.

Use Default Analytic Server. True or False specifies whether the default Analytic Server connection defined by an administrator in options.cfg is used (True), or if a different Analytic Server is used – the one specified in the stream/job (False).

URL. The URL for the Analytic Server in the format `https://hostname:port/contextroot`, where `hostname` is the IP address or host name of the Analytic Server, `port` is its port number, and `contextroot` is the context root of the Analytic Server.

Tenant. The name of the tenant that the SPSS® Modeler Server is a member of.

Credential. The credential for logging on to Analytic Server.

Service Principal Name. The Kerberos service principal name.

Config File Path. The Kerberos service config file path.

Data Source. The data source name on Analytic Server.

IBM® SPSS® Modeler Job Properties - Notifications

Allows you to specify email notifications for job step failure and success.

Click the Update button in each case to add or delete notification recipients.

Viewing Streams in IBM SPSS Modeler

You can launch the IBM® SPSS® Modeler application directly from IBM SPSS Deployment Manager.

To view the stream in the IBM SPSS Modeler application, double-click on the stream in the Content Explorer. The system launches the IBM SPSS Modeler application and displays the stream in IBM SPSS Modeler.

It is important to note that if any changes are made to the files in a job--for example, an IBM SPSS Modeler stream (.str)--any job that contains the file is affected. When changes are made to the file, a new version of the file is saved to the repository. However, the job that contains the file is not automatically updated with the modified file. To incorporate the file updates into the affected job:

1. Reopen the job. When the job is reopened, an asterisk appears with the job name in the job canvas, indicating that the job contains unsaved changes.
2. Resave the job.

Related information

- [Working with IBM SPSS Modeler streams](#)
 - [IBM SPSS Modeler Server configuration](#)
 - [Viewing IBM SPSS Modeler job properties](#)
 - [IBM SPSS Modeler Job Properties - General](#)
 - [IBM SPSS Modeler Job Properties - Data Files](#)
 - [Specifying the input file location](#)
 - [IBM SPSS Modeler Job Properties - ODBC data sources](#)
 - [IBM SPSS Modeler Job Properties - Parameters](#)
 - [IBM SPSS Modeler Job Properties - Results](#)
 - [Viewing output results](#)
 - [IBM SPSS Modeler Completion Codes](#)
 - [IBM SPSS Modeler Stream Limitations](#)
-

IBM SPSS Modeler Completion Codes

The completion codes for IBM® SPSS® Modeler jobs are described in the following table. Use these completion codes for any conditional relationships that involve IBM SPSS Modeler streams.

Table 1. Completion codes for IBM SPSS Modeler jobs

Code	Description
0	success
1	stream execution error
2	publishing error
8	unknown error

Related information

- [Working with IBM SPSS Modeler streams](#)
 - [IBM SPSS Modeler Server configuration](#)
 - [Viewing IBM SPSS Modeler job properties](#)
 - [IBM SPSS Modeler Job Properties - General](#)
 - [IBM SPSS Modeler Job Properties - Data Files](#)
 - [Specifying the input file location](#)
 - [IBM SPSS Modeler Job Properties - ODBC data sources](#)
 - [IBM SPSS Modeler Job Properties - Parameters](#)
 - [IBM SPSS Modeler Job Properties - Results](#)
 - [Viewing output results](#)
 - [Viewing Streams in IBM SPSS Modeler](#)
 - [IBM SPSS Modeler Stream Limitations](#)
-

IBM SPSS Modeler Stream Limitations

When you work with streams in the IBM® SPSS® Deployment Manager, the system has the following constraints:

- Naming. If the node name, label, and type are the same, you cannot schedule a job for execution because a conflict arises.
- Scripting. If a stream script contains overrides for a specific node—for example, the script sets the output location for a graph—the script supersedes any conflicting user-specified values defined in Deployment Manager.
- Supernodes. Execution of an IBM SPSS Modeler job step corresponds to the processing and execution of all top-level terminal nodes in the stream. If any terminal node is a terminal supernode, the terminal nodes within that supernode execute recursively. In contrast, if source or process supernodes, which are by definition non-terminal, contain terminal nodes, those nodes do not execute. The terminal nodes in the non-terminal supernodes do appear in Deployment Manager but are not processed during execution of the step.
- Parameters. Parameters defined for supernodes cannot have the same name as parameters defined for a stream. The names must be unique.

- Text Analytics node restrictions. You cannot use the SPSS Modeler Text Analytics File List or Web Feed nodes for scoring within an IBM SPSS Collaboration and Deployment Services - Scoring configuration.

In addition to the items listed above, there are the following specific constraints.

- No unlock checkbox is available when you store a stream in the IBM SPSS Collaboration and Deployment Services repository. Unlock is the default when storing streams. To lock or unlock an object, choose Tools > Repository > Explore, navigate to the object, and right-click on its name to display the context menu.
- In Deployment Manager, when you run a job that contains a stream with an Evaluation node set to produce a Gains graph, the graph output may be incomplete if the system is running under Oracle Weblogic 11g using the Oracle JRockit JRE. To avoid this problem, use the IBM JRE.

Node Types

When you open an IBM® SPSS® Modeler stream from IBM SPSS Deployment Manager, you see that the stream nodes are represented by different shaped icons. Circle icons represent source data nodes, while hexagonal nodes represent processing operations on data records and fields. Triangles indicate graphical output, while pentagons represent modeling nodes. Output (other than graphical) and export operations are indicated by rectangular nodes.

The various nodes are fully described in the *IBM SPSS Modeler Source, Process and Output Nodes* and the *IBM SPSS Modeler Modeling Nodes* guides.

Scoring Service

The Scoring Service allows client applications to employ real-time scores derived from predictive models developed in IBM® SPSS® Modeler. The service fetches the specified model, loads it, invokes the correct scoring implementation, and returns the result to the client.

Scoring is the process of generating real-time values by supplying predictive models with input data. A scoring model is any artifact that can be used to produce output values given input data. In general, to use a model for generating scores:

1. Select a model to use for scoring from the IBM SPSS Collaboration and Deployment Services Repository.
2. Define a scoring configuration for the model.
3. Supply the configured model with data and generate scores.

For more information about scoring and the Scoring Service, see the IBM SPSS Collaboration and Deployment Services documentation.

- [IBM SPSS Modeler stream limitations](#)

IBM SPSS Modeler stream limitations

When you work with streams in IBM® SPSS® Deployment Manager, the system has the following constraints for the Scoring Service:

- Supernodes. Source nodes inside supernodes are not supported, and terminal nodes inside supernodes are not supported.
- Geospatial nodes. Geospatial nodes are not supported.
- Model builder nodes. Model builder node as the terminal node is not supported.
- In-database mining. Streams containing in-database mining nuggets are not supported.
- UDF, UDA, and WUDA. Streams using database functions (UDF), database aggregations (UDA), or database window aggregates (WUDA) are not supported.
- Source nodes. Source nodes with an output data model containing a list type are not supported.
- Terminal nodes. Terminal nodes with an input data model containing a list type are not supported.
- Text Analytics node restrictions. You cannot use the SPSS Modeler Text Analytics File List or Web Feed nodes for scoring within an IBM SPSS Collaboration and Deployment Services - Scoring configuration.
- Analytic Server source nodes. If you have a job containing an SPSS Modeler stream that uses an Analytic Server source node, you must allow a direct connection between the IBM SPSS Collaboration and Deployment Services Server and Analytic Server. Otherwise, the job will fail if the firewall blocks the connection between the two servers.

Supported languages

IBM® SPSS® Modeler supports R and Apache Spark (via Python). See the following sections for more information.

- [R](#)
 - [Python for Spark](#)
 - [Extension nodes](#)
-

R

IBM® SPSS® Modeler supports R.

Allowable R syntax

- In the syntax field on the Syntax tab of the various Extension nodes, only statements and functions that are recognized by R are allowed.
- For the Extension Transform node and the Extension model nugget, data passes through the R script (in batch). For this reason, R scripts for model scoring and process nodes shouldn't include operations that span or combine rows in the data, such as sorting or aggregation. This limitation is imposed to ensure that data can be split up in a Hadoop environment, and during in-database mining. Extension Output and Extension model building nodes do not have this limitation.
- The addition of a non-batch data transfer mode, in both the Extension Transform node and the Extension model nugget, means that you can either span or combine rows in the data in SPSS Modeler Server.
- All R nodes can be seen as independent global R environments. Therefore, using `library` functions within the two separate R nodes requires the loading of the R library in both R scripts.
- To display the value of an R object that is defined in your R script, you must include a call to a printing function. For example, to display the value of an R object that is called `data`, include the following line in your R script:

```
print(data)
```

- You cannot include a call to the R `setwd` function in your R script because this function is used by IBM SPSS Modeler to control the file path of the R scripts output file.
 - Stream parameters that are defined for use in CLEM expressions and scripting are not recognized if used in R scripts.
 - IBM SPSS Modeler doesn't support the interactive plot in R
 - Some R objects are automatically populated when Extension nodes are used in a flow (for example, `modelerData` and `modelerDataModel` data frames). For more information, see [Model building syntax](#).
-

Python for Spark

IBM® SPSS® Modeler supports Python scripts for Apache Spark.

Note:

- Python nodes depend on the Spark environment.
 - Python scripts must use the Spark API because data will be presented in the form of a Spark DataFrame.
 - Old nodes created in version 17.1 will still only run against IBM SPSS Analytic Server (the data originates from an IBM SPSS Analytic Server source node and has not been extracted to IBM SPSS Modeler Server). New Python and Custom Dialog Builder nodes created in version 18.0 or later can run against IBM SPSS Modeler Server.
 - When installing Python, make sure all users have permission to access the Python installation.
 - If you want to use the Machine Learning Library (MLlib), you must install a version of Python that includes NumPy. Then you must configure the IBM SPSS Modeler Server (or the local server in IBM SPSS Modeler Client) to use your Python installation. For details, see [Scripting with Python for Spark](#).
 - [Scripting with Python for Spark](#)
 - [Switching to a different Python environment](#)
-

Scripting with Python for Spark

IBM® SPSS® Modeler can execute Python scripts using the Apache Spark framework to process data. This documentation provides the Python API description for the interfaces provided.

The IBM SPSS Modeler installation includes a Spark distribution (for example, IBM SPSS Modeler 18.4 includes Spark 3.2.1).

Prerequisites

- If you plan to execute Python/Spark scripts against IBM SPSS Analytic Server, you must have a connection to Analytic Server, and Analytic Server must have access to a compatible installation of Apache Spark. Refer to your IBM SPSS Analytic Server documentation for details

about using Apache Spark as the execution engine.

- If you plan to execute Python/Spark scripts against IBM SPSS Modeler Server (or the local server included with IBM SPSS Modeler Client, which requires Windows 64 or Mac64), you no longer need to install Python and edit options.cfg to use your Python installation. Starting with version 18.1, IBM SPSS Modeler now includes a Python distribution. However, if you require a certain module that is not included with the default IBM SPSS Modeler Python distribution, you can go to <Modeler_installation_directory>/python and install additional packages. Even though a Python distribution is now included with IBM SPSS Modeler, you can still point to your own Python installation as in previous releases if desired by adding the following option to options.cfg:

```
# Set to the full path to the python executable  
(including the executable name) to enable use of PySpark.  
eas_pyspark_python_path, ""
```

Windows example:

```
eas_pyspark_python_path, "C:\\Your_Python_Install\\python.exe"
```

Linux example:

```
eas_pyspark_python_path, "/Your_Python_Install/bin/python"
```

Note: If you point to your own Python installation, it must be version 3.8.x. IBM SPSS Modeler was tested with Anaconda for Python 3.8 and Python 3.8.6.

- If you plan to determine the PySpark version that is used by the IBM SPSS Modeler, you can run the following script.

```
import pkg_resources  
    print("pandas version")  
    print(pkg_resources.get_distribution("pandas").version)  
    print("pyspark version")  
    print(pkg_resources.get_distribution("pyspark").version)
```

Based on your OS, run the following commands in the corresponding paths before you run the script to determine the PySpark version.

For Windows

Path: <Modeler-InstallationDirectory>\18.4\spark\python
Command:

```
"<Modeler-InstallationDirectory>\18.4\python\python.exe" setup.py sdist
```

For Mac

Path: <Modeler-InstallationDirectory>/18.4/IBM SPSS Modeler.app/Contents/spark/python
Command:

```
"<Modeler-InstallationDirectory>/18.4/IBM SPSS Modeler.app/Contents/python/bin/python3" setup.py  
sdist
```

For Linux

Path: <Modeler-InstallationDirectory>/18.4/spark/python
Command:

```
<Modeler-InstallationDirectory>/18.4/python/bin/python3 setup.py sdist
```

Note: <Modeler-InstallationDirectory> corresponds to the IBM SPSS Modeler Thick Client installation for local server, whereas it corresponds to the IBM SPSS Modeler Server installation for remote server.

The IBM SPSS Analytic Server context object

The execution context for a Python/Spark script is defined by an Analytic Server context object. When running against IBM SPSS Modeler Server, the context object is for the embedded version of Analytic Server that is included with the IBM SPSS Modeler Server installation. To obtain the context object, the script must include the following:

```
import spss.pyspark.runtime  
asContext = spss.pyspark.runtime.getContext()
```

From the Analytic Server context, you can obtain the Spark context and the SQL context:

```
sparkContext = asc.getSparkContext()  
sqlContext = asc.getSparkSQLContext()
```

Refer to your Apache Spark documentation for information about the Spark context and the SQL context.

Accessing data

Data is transferred between a Python/Spark script and the execution context in the form of a Spark SQL DataFrame. A script that consumes data (that is, any node except a source node) must retrieve the data frame from the context:

```
inputData = asContext.getSparkInputData()
```

A script that produces data (that is, any node except a terminal node) must return a data frame to the context:

```
asContext.setSparkOutputData(outputData)
```

You can use the SQL context to create an output data frame from an RDD where required:

```
outputData = sqlContext.createDataFrame(rdd)
```

Defining the data model

A node that produces data must also define a data model that describes the fields visible downstream of the node. In Spark SQL terminology, the data model is the schema.

A Python/Spark script defines its output data model in the form of a `pyspark.sql.types.StructType` object. A `StructType` describes a row in the output data frame and is constructed from a list of `StructField` objects. Each `StructField` describes a single field in the output data model.

You can obtain the data model for the input data using the `:schema` attribute of the input data frame:

```
inputSchema = inputData.schema
```

Fields that are passed through unchanged can be copied from the input data model to the output data model. Fields that are new or modified in the output data model can be created using the `StructField` constructor:

```
field = StructField(name, dataType, nullable=True, metadata=None)
```

Refer to your Spark documentation for information about the constructor.

You must provide at least the field name and its data type. Optionally, you can specify metadata to provide a measure, role, and description for the field (see [Data metadata](#)).

DataModelOnly mode

IBM SPSS Modeler needs to know the output data model for a node, before the node is executed, in order to enable downstream editing. To obtain the output data model for a Python/Spark node, IBM SPSS Modeler executes the script in a special "data model only" mode where there is no data available. The script can identify this mode using the `isComputeDataModelOnly` method on the Analytic Server context object.

The script for a transformation node can follow this general pattern:

```
if asContext.isComputeDataModelOnly():
    inputSchema = asContext.getSparkInputSchema()
    outputSchema = ... # construct the output data model
    asContext.setSparkOutputSchema(outputSchema)
else:
    inputData = asContext.getSparkInputData()
    outputData = ... # construct the output data frame
    asContext.setSparkOutputData(outputData)
```

Building a model

A node that builds a model must return to the execution context some content that describes the model sufficiently that the node which applies the model can recreate it exactly at a later time.

Model content is defined in terms of key/value pairs where the meaning of the keys and the values is known only to the build and score nodes and is not interpreted by Modeler in any way. Optionally the node may assign a MIME type to a value with the intent that Modeler might display those values which have known types to the user in the model nugget.

A value in this context may be PMML, HTML, an image, etc. To add a value to the model content (in the build script):

```
asContext.setModelContentFromString(key, value, mimeType=None)
```

To retrieve a value from the model content (in the score script):

```
value = asContext.getModelContentToString(key)
```

As a shortcut, where a model or part of a model is stored to a file or folder in the file system you can bundle all the content stored to that location in one call (in the build script):

```
asContext.setModelContentFromPath(key, path)
```

Note that in this case there is no option to specify a MIME type because the bundle may contain various content types.

If you need a temporary location to store the content while building the model you can obtain an appropriate location from the context:

```
path = asContext.createTemporaryFolder()
```

To retrieve existing content to a temporary location in the file system (in the score script):

```
path = asContext.getModelContentToPath(key)
```

Error handling

To raise errors, throw an exception from the script and display it to the IBM SPSS Modeler user. Some exceptions are predefined in the module `spss.pyspark.exceptions`. For example:

```
from spss.pyspark.exceptions import ASContextException
if ... some error condition ...
    raise ASContextException("message to display to user")
```

- [Analytic Server Context](#)
 - [Data metadata](#)
 - [Date, time, timestamp](#)
 - [Exceptions](#)
 - [Examples](#)
-

Analytic Server Context

The Context provides support for the Analytic Server context interface for interaction with IBM® SPSS® Analytic Server.

AnalyticServerContext Objects

`AnalyticServerContext` Objects set up the context environment which provides several interfaces for interacting with IBM SPSS Analytic Server. An application that wants to construct this context instance must do so using the `spss.pyspark.runtime.getContext()` interface rather than implementing the interface directly.

Returns the Pyspark python `SparkContext` instance:

```
ctx.getSparkContext() : SparkContext
```

Returns the Pyspark python `SQLContext` instance:

```
ctx.getSparkSQLContext() : SQLContext
```

Returns `True` to describe whether the execution is made only to compute the output data model. Otherwise returns `False`:

```
ctx.isComputeDataModelOnly() : Boolean
```

Returns `True` if the script is running in the Spark environment. Currently, it always returns `True`:

```
ctx.isSparkExecution() : Boolean
```

Loads input data from the upstream temporary file and generates the `pyspark.sql.DataFrame` instance:

```
ctx.getSparkInputData() : DataFrame
```

Returns a `pyspark.sql.StructType` instance generated from the input data model. Returns `None` if the input data model does not exist:

```
ctx.getSparkInputSchema() : StructType
```

Serializes the output data frame into Analytic Server context and returns the context:

```
ctx.setSparkOutputData( outDF ) : AnalyticServerContext
```

Parameter:

- `outDF (DataFrame) : The output data frame value`

Exceptions:

- `DataOutputNotSupported` : If this interface is invoked in the function `pyspark:buildmodel`
- `ASContextException` : If the output data frame is `None`
- `InconsistentOutputDataModel` : The field names and storage type information common to both objects is inconsistent

Converts the `outSchema StructType` instance into a data model, serializes it into the Analytic Server context, and returns the context:

```
ctx.setSparkOutputSchema(outSchema) : AnalyticServerContext
```

Parameter:

- `outSchema(StructType)` : The output `StructType` object

Exceptions:

- `ASContextException` : If the output schema instance is `None`
- `InconsistentOutputDataModel` : The field names and storage type information common to both objects is inconsistent

Stores the location of model building output to the Analytic Server context and returns the context:

```
ctx.setModelContentFromPath(key, path, mimetype=None) : AnalyticServerContext
```

The path can be a directory path which should use the `ctx.createTemporaryFolder()` API to generate , when everything under the directory is packaged up as model content.

Parameters:

- `key (string)` : key string value
- `path (string)` : location of model building output string path
- `mimetype (string, optional)` : the MIME type of the content

Exceptions:

- `ModelErrorNotSupported` : When not invoking this API from the `pyspark:buildmodel` function
- `KeyError` : If the key attribute is `None` or the string is empty

Stores the model building content, metadata, or other attributes to the Analytic Server context and returns the context:

```
ctx.setModelContentFromString(key, value, mimetype=None) : AnalyticServerContext
```

Parameters:

- `key (string)` : key string value
- `value (string)` : the model metadata string value
- `mimetype (string, optional)` : the MIME type of the content

Exceptions:

- `ModelErrorNotSupported` : When not invoking this API from the `pyspark:buildmodel` function
- `KeyError` : If the key attribute is `None` or the string is empty

Returns the temporary folder location that is managed by Analytic Server; this can be used to store the model content:

```
ctx.createTemporaryFolder() : string
```

Exception:

- `ModelErrorNotSupported` : When not invoking this API from the `pyspark:buildmodel` function

Returns the location of the model which matches the input key:

```
ctx.getModelContentToPath(key) : string
```

Parameter:

- `key (string)` : key string value

Exceptions:

- `ModelErrorNotSupported` : When not invoking this API from the `pyspark:applymodel` function
- `KeyError` : If the key attribute is `None` or the string is empty
- `IncompatibleModelContentType` : If the model content type is not a container

Returns the model content, metadata of the model, or other model attributes which match the input key:

```
ctx.getModelContentToString(key) : string
```

Parameter:

- `key (string)` : key string value

Exceptions:

- `ModelErrorNotSupported` : When not invoking this API from the `pyspark:applymodel` function
- `KeyError` : If the key attribute is `None`, or the string is empty, or the key does not exist
- `IncompatibleModelContentType` : If the model content type is not consistent

Returns the mime-type assigned to the input key. It returns `None` if the specified content has no mime type:

```
ctx.getModelContentMimeType(key) : string
```

Parameter:

- `key (string)` : key string value

Exceptions:

- `ModelErrorNotSupported` : When not invoking this API from the `pyspark:applymodel` function
- `KeyError` : If the key attribute is `None`, or the string is empty, or the key does not exist

Data metadata

This section describes how to set up the data model attributes based on `pyspark.sql.StructField`.

`spss.datamodel.Role Objects`

This class enumerates valid roles for each field in a data model.

`BOTH`: Indicates that this field can be either an antecedent or a consequent.

`FREQWEIGHT`: Indicates that this field is used to be as frequency weight; this is not displayed to the user.

`INPUT`: Indicates that this field is a predictor or an antecedent.

`NONE`: Indicates that this field is not used directly during modeling.

`TARGET`: Indicates that this field is predicted or a consequent.

`PARTITION`: Indicates that this field is used to identify the data partition.

`RECORDID`: Indicates that this field is used to identify the record id.

`SPLIT`: Indicates that this field is used to split the data.

`spss.datamodel.Measure Objects`

This class enumerates measurement levels for fields in a data model.

`UNKNOWN`: Indicates that the measure type is unknown.

`CONTINUOUS`: Indicates that the measure type is continuous.

`NOMINAL`: Indicates that the measure type is nominal.

`FLAG`: Indicates that the field value is one of two values.

`DISCRETE`: Indicates that the field value should be interpreted as a collection of values.

`ORDINAL`: Indicates that the measure type is ordinal.

`TYPELESS`: Indicates that the field can have any value compatible with its storage.

`pyspark.sql.StructField Objects`

Represents a field in a `StructType`. A `StructField` object comprises four fields:

- `name (string)`: name of a `StructField`
- `dataType (pyspark.sql.DataType)`: specific data type
- `nullable (bool)`: if the values of a `StructField` can contain `None` values
- `metadata (dictionary)`: a python dictionary used to store the option attributes

You can use the metadata dictionary instance to store the measure, role, or label attribute for the specific field. The key words for these attributes are:

- `measure`: the key word for `measure` attribute
- `role`: the key word for `role` attribute
- `displayLabel`: the key word for `label` attribute

Example:

```
from spss.datamodel.Role import Role
from spss.datamodel.Measure import Measure
_metadata = {}
_metadata['measure'] = Measure.TYPELESS
_metadata['role'] = Role.NONE
_metadata['displayLabel'] = "field label description"
StructField("userName", StringType(), nullable=False,
metadata=_metadata)
```

Date, time, timestamp

For operations that use date, time, or timestamp type data, the value is converted to the real value based on the value **1970-01-01:00:00:00** (using Coordinated Universal Time).

For the date, the value represents the number of days, based on the value **1970-01-01** (using Coordinated Universal Time).

For the time, the value represents the number of seconds at 24 hours.

For the timestamp, the value represents the number of seconds based on the value **1970-01-01:00:00:00** (using Coordinated Universal Time).

Exceptions

This section describes possible exception instances.

MetadataException Objects

A subclass of python Exception.

This exception is thrown if an error occurs during operate the metadata object.

UnsupportedOperationException Objects

A subclass of python Exception.

This exception is thrown if the specific operation does not allow execution.

InconsistentOutputDataModel Objects

A subclass of python Exception.

This Exception is thrown if both `setSparkOutputSchema` and `setSparkOutputData` are invoked but the field names and storage type information common to both objects are inconsistent.

IncompatibleModelContentType Objects

A subclass of python Exception.

This Exception is thrown during the following scenarios:

- Using `setModelContentFormString` to set model but using `getModelContentToPath` to get value
- Using `setModelContentFormPath` to set model but using `getModelContentToString` to get value

DataOutputNotSupportedException Objects

A subclass of python Exception.

This exception is raised in `setSparkOutputData` in an execution handled by function `pyspark:buildmodel`.

ModelErrorNotSupportedException Objects

A subclass of python Exception.

This exception is only raised if the script does not invoke the `getModelContentPathByKey` and `getModelContentToString` API in the `pyspark:applymodel` function.

ModelOutputNotSupportedException Objects

A subclass of python Exception.

This exception is only raised if the script does not invoke the `setModelContentFromPath` and `setModelContentFromString` APIs in the `pyspark:buildmodel` function.

ASContextException Objects

A subclass of python Exception.

This exception is thrown if an unexpected runtime exception occurs.

Examples

This section contains Python for Spark scripting examples.

Basic scripting example for processing data

```
import spss.pyspark.runtime
from pyspark.sql.types import *

ctx = spss.pyspark.runtime.getContext()

if ctx.isComputeDataModelOnly():
    _schema = ctx.getSparkInputSchema()
    ctx.setSparkOutputSchema(_schema)
else:
    _structType = ctx.getSparkInputSchema()
    df = ctx.getSparkInputData()
    _newDF = df.sample(False, 0.01, 1)
    ctx.setSparkOutputData(_newDF)
```

Example model building script, using the LinearRegressionWithSGD algorithm

```
from pyspark.context import SparkContext
from pyspark.sql.context import SQLContext
from pyspark.sql import Row
from pyspark.mllib.regression import
LabeledPoint,LinearRegressionWithSGD, LinearRegressionModel
from pyspark.mllib.linalg import DenseVector
import numpy
import json

import spss.pyspark.runtime
from spss.pyspark.exceptions import ASContextException

ascontext = spss.pyspark.runtime.getContext()
sc = ascontext.getSparkContext()
df = ascontext.getSparkInputData()

# field settings amd algorithm parameters

target = '%%target_field%%'
predictors = [%%predictor_fields%%]
num_iterations=%%num_iterations%%
prediction_field = "$LR-" + target

# save linear regression model to a filesystem path

def save(model, sc, path):
    data =
    sc.parallelize([json.dumps({"intercept":model.intercept,"weights":model.weights.tolist()})])
    data.saveAsTextFile(path)

# print model details to stdout
```

```

def dump(model,predictors):
    print(prediction_field+" = " + str(model.intercept))
    weights = model.weights.tolist()
    for i in range(0,len(predictors)):
        print("\t"+predictors[i]+"*"+ str(weights[i]))

# check that required fields exist in the input data

input_field_names = [ty[0] for ty in df.dtypes[:]]
if target not in input_field_names:
    raise ASContextException("target field "+target+" not found") for predictor in predictors:
    if predictor not in input_field_names:
        raise ASContextException("predictor field "+predictor+" not found")

# define map function to convert from dataframe Row objects to mllib LabeledPoint

def row2LabeledPoint(target,predictors,row):
    pvals = []
    for predictor in predictors:
        pval = getattr(row,predictor)
        pvals.append(float(pval))
    tval = getattr(row,target)
    return LabeledPoint(float(tval),DenseVector(pvals))

# convert dataframe to an RDD containing LabeledPoint

training_points = df.rdd.map(lambda row:
row2LabeledPoint(target,predictors,row))

# build the model

model = LinearRegressionWithSGD.train(training_points,num_iterations,intercept=True)

# write a text description of the model to stdout

dump(model,predictors)

# save the model to the filesystem and store into the output model content

modelpath = ascontext.createTemporaryFolder()
save(model,sc,modelpath)
ascontext.setModelContentFromPath("model",modelpath)

```

Example model scoring script, using the LinearRegressionWithSGD algorithm

```

import json
import spss.pyspark.runtime
from pyspark.sql import Row
from pyspark.mllib.regression import
LabeledPoint,LinearRegressionWithSGD, LinearRegressionModel
from pyspark.mllib.linalg import DenseVector
from pyspark.sql.context import SQLContext
import numpy
from pyspark.sql.types import DoubleType, StructField

ascontext = spss.pyspark.runtime.getContext()
sc = ascontext.getSparkContext()

prediction_field = "$LR-" + '%%target_field%%'
predictors = [%%predictor_fields%%]

# compute the output schema by adding the prediction field
outputSchema = ascontext.getSparkInputSchema()
outputSchema.fields.append(StructField(prediction_field,
DoubleType(), nullable=True))

# make a prediction based on a regression model and Dataframe Row object
# return a list containing the input row values and the predicted value
def predict(row,model,predictors,infields,prediction_field_name):
    pvals = []
    rdict = row.asDict()
    for predictor in predictors:
        pvals.append(float(rdict[predictor]))
    estimate = float(model.predict(pvals))
    result = []
    for field in infields:
        result.append(rdict[field])
    result.append(estimate)
    return result

```

```

# load a serialized model from the filesystem

def load(sc, path):
    js = sc.textFile(path).take(1)[0]
    obj = json.loads(js)
    weights = numpy.array(obj["weights"])
    intercept = obj["intercept"]
    return LinearRegressionModel(weights, intercept)

ascontext.setSparkOutputSchema(outputSchema)

if not ascontext.isComputeDataModelOnly():
    # score the data in the input data frame
    indf = ascontext.getSparkInputData()

    model_path = ascontext.getModelContentToPath("model")
    model = load(sc, model_path)

    # compute the scores
    infield_names = [ty[0] for ty in indf.dtypes[:]]
    scores_rdd = indf.rdd.map(lambda row:predict(row,model,predictors,infield_names,prediction_field))

    # create an output DataFrame containing the scores
    sqlCtx = SQLContext(sc)
    outdf = sqlCtx.createDataFrame(scores_rdd,schema=outputSchema)

    # return the output DataFrame as the result
    ascontext.setSparkOutputData(outdf)

```

Switching to a different Python environment

You can change the Python environment used by the Extension nodes and Custom Dialog Builder in IBM® SPSS® Modeler:

1. Grant edit permissions for options.cfg:
 - a. From the IBM SPSS Modeler Server installation directory, open config/options.cfg in a text editor.
 - b. Modify the following line to grant users edit permissions:

```
administrators, "*"
```

 - c. Save the file and then restart the IBM SPSS Modeler Server.
2. Open IBM SPSS Modeler and connect to the IBM SPSS Modeler Server.
3. Go to Tools>Manage Modeler Server Configuration.
4. Under Data File Access, modify the Python executable path and then click OK.
5. Restart the IBM SPSS Modeler Server and then reconnect to it in IBM SPSS Modeler.
6. Go to Tools>Manage Modeler Server Configuration and verify that your new Python path was saved.

Extension nodes

To complement IBM® SPSS® Modeler and its data mining abilities, the Extension nodes enable expert users to input their own R scripts or Python for Spark scripts to carry out data processing, model building, and model scoring.

- [Extension Export node](#)
- [Extension Output node](#)
- [Extension Model node](#)
- [Extension model nugget](#)
- [Extension Transform node](#)
- [Extension Import node](#)

Extension Export node

With the Extension Export node, you can run R or Python for Spark scripts to export data.

- [Extension Export node - Syntax tab](#)
- [Extension Export node - Console Output tab](#)
- [Publishing streams](#)

Extension Export node - Syntax tab

Select your type of syntax – R or Python for Spark. See the following sections for more information. When your syntax is ready, you can click Run to execute the Extension Export node.

R Syntax

R Syntax. You can enter, or paste, custom R scripting syntax for data analysis into this field.

Convert flag fields. Specifies how flag fields are treated. There are two options: Strings to factor, Integers and Reals to double, and Logical values (True, False). If you select Logical values (True, False) the original values of the flag fields are lost. For example, if a field has values Male and Female, these are changed to True and False.

Convert missing values to the R 'not available' value (NA). When selected, any missing values are converted to the R NA value. The value NA is used by R to identify missing values. Some R functions that you use might have an argument that can be used to control how the function behaves when the data contain NA. For example, the function might allow you to choose to automatically exclude records that contain NA. If this option is not selected, any missing values are passed to R unchanged, and might cause errors when your R script is executed.

Convert date/time fields to R classes with special control for time zones. When selected, variables with date or datetime formats are converted to R date/time objects. You must select one of the following options:

- **R POSIXct.** Variables with date or datetime formats are converted to R POSIXct objects.
- **R POSIXlt (list).** Variables with date or datetime formats are converted to R POSIXlt objects.

Note: The POSIX formats are advanced options. Use these options only if your R script specifies that datetime fields are treated in ways that require these formats. The POSIX formats do not apply to variables with time formats.

Python Syntax

Python Syntax. You can enter, or paste, custom Python scripting syntax for data analysis into this field. For more information about Python for Spark, see [Python for Spark](#) and [Scripting with Python for Spark](#).

Extension Export node - Console Output tab

The Console Output tab contains any output that is received when the R script or Python for Spark script on the Syntax tab runs (for example, if using an R script, it shows output received from the R console when the R script in the R Syntax field on the Syntax tab is executed). This output might include R or Python error messages or warnings that are produced when the R or Python script is executed. The output can be used, primarily, to debug the script. The Console Output tab also contains the script from the R Syntax or Python Syntax field.

Every time the Extension Export script is executed, the content of the Console Output tab is overwritten with the output received from the R console or Python for Spark. The output cannot be edited.

Publishing streams

Publishing streams is done directly from IBM® SPSS® Modeler using any of the standard export nodes: Database, Flat File, Statistics Export, Extension Export, Data Collection Export, SAS Export, Excel, and XML Export nodes. The type of export node determines the format of the results to be written each time the published stream is executed using the IBM SPSS Modeler Solution Publisher Runtime or external application. For example, if you want to write your results to a database each time the published stream is run, use a Database export node.

To publish a stream

1. Open or build a stream in the normal manner and attach an export node at the end.
2. On the Publish tab in the export node, specify a root name for the published files (that is, the file name to which the .pim, .par, and .xml extensions will be appended).
3. Click Publish to publish the stream, or select Publish the stream to automatically publish the stream each time the node is executed.

Published name. Specify the root name for the published image and parameter files.

- The image file (*.pim) provides all of the information needed for the Runtime to execute the published stream exactly as it was at the time of export. If you are confident that you will not need to change any of the settings for the stream (such as the input data source or the output data file), you can deploy the image file only.

- The parameter file (*.par) contains configurable information about data sources, output files, and execution options. If you want to be able to control the input or output of the stream without republishing the stream, you will need the parameter file as well as the image file.
 - The metadata file (*.xml) describes the inputs and outputs of the image and their data models. It is designed for use by applications which embed the runtime library and which need to know the structure of the input and output data.
- Note: This file is only produced if you select the Publish metadata option.

Publish parameters. If required, you can include stream parameters in the *.par file. You can change these stream parameter values when you execute the image either by editing the *.par file or through the runtime API.

This option enables the Parameters button. The Publish Parameters dialog box is displayed when you click the button.

Choose the parameters you want to include in the published image by selecting the relevant option in the Publish column.

On stream execution. Specifies whether the stream is automatically published when the node is executed.

- Export data. Executes the export node in the standard manner, without publishing the stream. (Basically, the node executes in IBM SPSS Modeler the same way it would if IBM SPSS Modeler Solution Publisher were not available.) If you select this option, the stream will not be published unless you do so explicitly by clicking Publish in the export node dialog box. Alternatively, you can publish the current stream using the Publish tool on the toolbar or by using a script.
- Publish the stream. Publishes the stream for deployment using IBM SPSS Modeler Solution Publisher. Select this option if you want to automatically publish the stream every time it is executed.

Note:

- If you plan to run the published stream with new or updated data, it is important to note that the order of fields in the input file must be the same as the order of fields in the source node input file specified in the published stream.
- When publishing to external applications, consider filtering extraneous fields or renaming fields to conform with input requirements. Both can be accomplished using a Filter node prior to the export node.

Extension Output node

If Output to screen is selected on the Output tab of the Extension Output node dialog, on-screen output is displayed in an output browser window. The output is also added to the Output manager. The output browser window has its own set of menus that allow you to print or save the output, or export it to another format. The Edit menu only contains the Copy option. The Extension Output node's output browser has two tabs; the Text Output tab that displays text output, and the Graph Output tab that displays graphs and charts.

If Output to file is selected on the Output tab of the Extension Output node dialog, the output browser window is not displayed upon successful execution of the Extension Output node.

- [Extension Output node - Syntax tab](#)
- [Extension Output node - Console Output tab](#)
- [Extension Output node - Output tab](#)
- [Extension Output Browser](#)

Extension Output node - Syntax tab

Select your type of syntax – R or Python for Spark. See the following sections for more information. When your syntax is ready, you can click Run to execute the Extension Output node. The output objects are added to the Output manager, or optionally to the file specified in the Filename field on the Output tab.

R Syntax

R Syntax. You can enter, or paste, custom R scripting syntax for data analysis into this field.

Convert flag fields. Specifies how flag fields are treated. There are two options: Strings to factor, Integers and Reals to double, and Logical values (True, False). If you select Logical values (True, False) the original values of the flag fields are lost. For example, if a field has values Male and Female, these are changed to True and False.

Convert missing values to the R 'not available' value (NA). When selected, any missing values are converted to the R NA value. The value NA is used by R to identify missing values. Some R functions that you use might have an argument that can be used to control how the function behaves when the data contain NA. For example, the function might allow you to choose to automatically exclude records that contain NA. If this option is not selected, any missing values are passed to R unchanged, and might cause errors when your R script is executed.

Convert date/time fields to R classes with special control for time zones. When selected, variables with date or datetime formats are converted to R date/time objects. You must select one of the following options:

- **R POSIXct**. Variables with date or datetime formats are converted to R `POSIXct` objects.
- **R POSIXlt (list)**. Variables with date or datetime formats are converted to R `POSIXlt` objects.

Note: The POSIX formats are advanced options. Use these options only if your R script specifies that datetime fields are treated in ways that require these formats. The POSIX formats do not apply to variables with time formats.

Python Syntax

Python Syntax. You can enter, or paste, custom Python scripting syntax for data analysis into this field. For more information about Python for Spark, see [Python for Spark](#) and [Scripting with Python for Spark](#).

Extension Output node - Console Output tab

The Console Output tab contains any output that is received when the R script or Python for Spark script on the Syntax tab runs (for example, if using an R script, it shows output received from the R console when the R script in the R Syntax field on the Syntax tab is executed). This output might include R or Python error messages or warnings that are produced when the R or Python script is executed. The output can be used, primarily, to debug the script. The Console Output tab also contains the script from the R Syntax or Python Syntax field.

Every time the Extension Output script is executed, the content of the Console Output tab is overwritten with the output received from the R console or Python for Spark. The output cannot be edited.

Extension Output node - Output tab

Output name. Specifies the name of the output that is produced when the node is executed. When Auto is selected, the name of the output is automatically set to "R Output" or "Python Output" depending on the script type. Optionally, you can select Custom to specify a different name.

Output to screen. Select this option to generate and display the output in a new window. The output is also added to the Output manager.

Output to file. Select this option to save the output to a file. Doing so enables the Output Graph and Output File radio buttons.

Output Graph. Only enabled if Output to file is selected. Select this option to save any graphs that result from executing the Extension Output node to a file. Specify a filename to use for the generated output in the Filename field. Click the ellipses button (...) to choose a specific file and location. Specify the file type in the File type drop-down list. The following file types are available:

- Output object (.cou)
- HTML (.html)

Output Text. Only enabled if Output to file is selected. Select this option to save any text output that results from executing the Extension Output node to a file. Specify a filename to use for the generated output in the Filename field. Click the ellipses button (...) to specify a specific file and location. Specify the file type in the File type drop-down list. The following file types are available:

- HTML (.html)
- Output object (.cou)
- Text document (.txt)

Extension Output Browser

If Output to screen is selected on the Output tab of the Extension Output node dialog box, on-screen output is displayed in an output browser window. The output is also added to the Output manager. The output browser window has its own set of menus that allow you to print or save the output, or export it to another format. The Edit menu only contains the Copy option. The output browser of the Extension Output node has two tabs:

- The Text Output tab displays text output
- The Graph Output tab displays graphs and charts

If Output to file is selected on the Output tab of the Extension Output node dialog box instead of Output to screen, the output browser window is not displayed upon successful execution of the Extension Output node.

- [Extension Output Browser - Text Output tab](#)
- [Extension Output Browser - Graph Output tab](#)

Extension Output Browser - Text Output tab

The Text Output tab displays any text output that is generated when the R script or Python for Spark script on the Syntax tab of the Extension Output node is executed.

Note: R or Python for Spark error messages or warnings that result from executing your Extension Output script are always displayed on the Console Output tab of the Extension Output node.

Extension Output Browser - Graph Output tab

The Graph Output tab displays any graphs or charts that are generated when the R script or Python for Spark script on the Syntax tab of the Extension Output node is executed. For example, if your R script contains a call to the R `plot` function, the resulting graph is displayed on this tab.

Extension Model node

With the Extension Model node, you can run R or Python for Spark scripts to build and score models.

- [Extension Model node - Syntax tab](#)
- [Extension Model node - Model Options tab](#)
- [Extension Model node - Console Output tab](#)
- [Extension Model node - Text Output tab](#)

Extension Model node - Syntax tab

Select your type of syntax – R or Python for Spark. Then enter or paste your custom scripting syntax into one of the following fields. When your syntax is ready, you can click Run to execute the Extension Model node.

R syntax

R model building syntax. You can enter, or paste, custom R scripting syntax for model building into this field.

R model scoring syntax. You can enter, or paste, custom R scripting syntax for model scoring into this field.

Python for Spark syntax

Python model building syntax. You can enter, or paste, custom Python scripting syntax for model building into this field.

Python model scoring syntax. You can enter, or paste, custom Python scripting syntax for model scoring into this field.

For more information about Python for Spark, see [Python for Spark](#) and [Scripting with Python for Spark](#).

Extension Model node - Model Options tab

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified), or specify a custom name.

Extension Model node - Console Output tab

The Console Output tab contains any output that is received when the R script or Python for Spark script on the Syntax tab runs (for example, if using an R script, it shows output received from the R console when the R script in the R Syntax field on the Syntax tab is executed). This output might include R or Python error messages or warnings that are produced when the R or Python script is executed. The output can be used, primarily, to debug the script. The Console Output tab also contains the script from the R Syntax or Python Syntax field.

Every time the Extension Model script is executed, the content of the Console Output tab is overwritten with the output received from the R console or Python for Spark. The output cannot be edited.

Extension Model node - Text Output tab

The Text Output tab is present in the Extension Model node if requested by selecting the Display R text output check box on the Model Options tab of the Extension Model node dialog box. This tab can display only text output. Any text output that is produced by executing your R model building script is displayed on this tab. If you execute the model building script again, without having first specified a different name for the model, the content of the Text Output tab from the previous execution will be overwritten. The text output cannot be edited.

If you include a call to the R `sink` function in your script, any output that is produced after this function is saved to the specified file and is not displayed on the Text Output tab.

Note: R or Python for Spark error messages or warnings that result from executing your model building script are always displayed on the Console Output tab of the Extension Model node.

Extension model nugget

The Extension model nugget is generated and placed on the Models palette after executing the Extension Model node, which contains the R or Python for Spark script that defines the model building and model scoring. By default, the Extension model nugget contains the script that is used for model scoring, options for reading the data, and any output from the R console or Python for Spark. Optionally, the Extension model nugget can also contain various other forms of model output, such as graphs and text output. After the Extension model nugget is generated and added to the stream canvas, an output node can be connected to it. The output node is then used in the usual way within IBM® SPSS® Modeler streams for obtaining information about the data and models, and for exporting data in various formats.

To use this node with R, you must install IBM SPSS Modeler - Essentials for R. See the *IBM SPSS Modeler - Essentials for R: Installation Instructions* for installation instructions and compatibility information. You must also have a compatible version of R installed on your computer.

- [Extension model nugget - Syntax tab](#)
- [Extension model nugget - Model Options tab](#)
- [Extension model nugget - Graph Output tab](#)
- [Extension model nugget - Text Output tab](#)
- [Extension model nugget - Console Output tab](#)

Extension model nugget - Syntax tab

The Syntax tab is always present in the Extension model nugget.

R model scoring syntax. If using R, the R script that is used for model scoring is displayed in this field. By default, this field is enabled but not editable. To edit the R model scoring script, click Edit.

Python model scoring syntax. If using Python for Spark, the Python script that is used for model scoring is displayed in this field. By default, this field is enabled but not editable. To edit the Python model scoring script, click Edit.

Edit. Click Edit to make the scoring syntax field editable. You can then edit your model scoring script by typing in the scoring syntax field. For example, you might want to edit your model scoring script if you identify an error in your model scoring script after you have executed the Extension model nugget. Any changes that you make to the model scoring script in the Extension model nugget will be lost if you regenerate the model by executing the Extension Model node.

Extension model nugget - Model Options tab

The Model Options tab is always present in the Extension model nugget.

Read Data Options. These options only apply to R, not Python for Spark. With these options, you can specify how missing values, flag fields, and variables with date or datetime formats are handled.

- Read data in batches. If you are processing a large amount of data (that is too big to fit into the R engine's memory, for example), use this option to break the data down into batches that can be sent and processed individually. Specify the maximum number of data records to be included in each batch.

For both the Extension Transform node and the Extension Scoring nugget, data passes through the R script (in batch). For this reason, scripts for model scoring and process nodes that are run in either a Hadoop or Database environment should not include operations that span or combine rows in the data, such as sorting or aggregation. This limitation is imposed to ensure that data can be split up in a Hadoop environment, and during in-database mining. This limitation does not apply if the scripts for model scoring are run in SPSS® Modeler Server. Extension Output and Extension Model nodes do not have this limitation.

- Convert flag fields. Specifies how flag fields are treated. There are two options: Strings to factor, Integers and Reals to double, and Logical values (True, False). If you select Logical values (True, False) the original values of the flag fields are lost. For example, if a field has values Male and Female, these are changed to True and False.
- Convert missing values to the R 'not available' value (NA). When selected, any missing values are converted to the R NA value. The value NA is used by R to identify missing values. Some R functions that you use might have an argument that can be used to control how the function behaves when the data contain NA. For example, the function might allow you to choose to automatically exclude records that contain NA. If this option is not selected, any missing values are passed to R unchanged, and might cause errors when your R script is executed.
- Convert date/time fields to R classes with special control for time zones When selected, variables with date or datetime formats are converted to R date/time objects. You must select one of the following options:
 - **R POSIXct**. Variables with date or datetime formats are converted to R POSIXct objects.
 - **R POSIXlt (list)**. Variables with date or datetime formats are converted to R POSIXlt objects.

Note: The POSIX formats are advanced options. Use these options only if your R script specifies that datetime fields are treated in ways that require these formats. The POSIX formats do not apply to variables with time formats.

The options that are selected for the Convert flag fields, Convert missing values to the R 'not available' value (NA), and Convert date/time fields to R classes with special control for time zones controls are not recognized when the Extension model nugget is run against a database. When the node is run against a database, the default values for these controls are used instead:

- Convert flag fields is set to Strings to factor, Integers and Reals to double.
- Convert missing values to the R 'not available' value (NA) is selected.
- Convert date/time fields to R classes with special control for time zones is not selected.

Extension model nugget - Graph Output tab

The Graph Output tab is present in the Extension model nugget if requested by selecting the Display R graphs as HTML check box on the Model Options tab of the Extension Model node dialog box. Graphs that result from executing the model building R script can be displayed on this tab. For example, if your R script contains a call to the R `plot` function, the resulting graph is displayed on this tab. If you execute the model building script again, without having first specified a different name for the model, the content of the Graph Output tab from the previous execution will be overwritten.

Extension model nugget - Text Output tab

The Text Output tab is present in the Extension model nugget if requested by selecting the Display R text output check box on the Model Options tab of the Extension Model node dialog box. This tab can display only text output. Any text output that is produced by executing your Extension Model script is displayed on this tab. If you execute the Extension Model script again, without having first specified a different name for the model, the content of the Text Output tab from the previous execution will be overwritten. The text output cannot be edited.

Note:

- If you include a call to the R `sink` function in your script, any output that is produced after this function is saved to the specified file and is not displayed on the Text Output tab.
- Error messages or warnings that result from executing your Extension Model script are always displayed on the Console Output tab of the Extension Model node.

Extension model nugget - Console Output tab

The Console Output tab is always present in the Extension model nugget. It contains any output that is received when the R script or Python for Spark script on the Syntax tab runs (for example, if using an R script, it shows output received from the R console when the R script in the R model scoring syntax field on the Syntax tab of the Extension model nugget is executed). This output includes any R or Python error messages or warnings that are produced when the R or Python script is executed, and any text output from the R console. The output can be used, primarily, to debug the script.

Every time the model scoring script is executed, the content of the Console Output tab is overwritten with the output received from the R console or Python for Spark. The console output cannot be edited.

Extension Transform node

With the Extension Transform node, you can take data from an IBM® SPSS® Modeler stream and apply transformations to the data using R scripting or Python for Spark scripting. When the data has been modified, it is returned to the stream for further processing, model building and model scoring. The Extension Transform node makes it possible to transform data using algorithms that are written in R or Python for Spark, and enables the user to develop data transformation methods that are tailored to a particular problem.

To use this node with R, you must install IBM SPSS Modeler - Essentials for R. See the *IBM SPSS Modeler - Essentials for R: Installation Instructions* for installation instructions and compatibility information. You must also have a compatible version of R installed on your computer.

- [Extension Transform node - Syntax tab](#)
- [Extension Transform node - Console Output tab](#)

Extension Transform node - Syntax tab

Select your type of syntax – R or Python for Spark. See the following sections for more information. When your syntax is ready, you can click Run to execute the Extension Transform node.

R Syntax

R Syntax. You can enter, or paste, custom R scripting syntax for data analysis into this field.

Convert flag fields. Specifies how flag fields are treated. There are two options: Strings to factor, Integers and Reals to double, and Logical values (True, False). If you select Logical values (True, False) the original values of the flag fields are lost. For example, if a field has values Male and Female, these are changed to True and False.

Convert missing values to the R 'not available' value (NA). When selected, any missing values are converted to the R NA value. The value NA is used by R to identify missing values. Some R functions that you use might have an argument that can be used to control how the function behaves when the data contain NA. For example, the function might allow you to choose to automatically exclude records that contain NA. If this option is not selected, any missing values are passed to R unchanged, and might cause errors when your R script is executed.

Convert date/time fields to R classes with special control for time zones. When selected, variables with date or datetime formats are converted to R date/time objects. You must select one of the following options:

- **R POSIXct.** Variables with date or datetime formats are converted to R POSIXct objects.
- **R POSIXlt (list).** Variables with date or datetime formats are converted to R POSIXlt objects.

Note: The POSIX formats are advanced options. Use these options only if your R script specifies that datetime fields are treated in ways that require these formats. The POSIX formats do not apply to variables with time formats.

Python Syntax

Python Syntax. You can enter, or paste, custom Python scripting syntax for data analysis into this field. For more information about Python for Spark, see [Python for Spark](#) and [Scripting with Python for Spark](#).

Extension Transform node - Console Output tab

The Console Output tab contains any output that is received when the R script or Python for Spark script on the Syntax tab runs (for example, if using an R script, it shows output received from the R console when the R script in the R Syntax field on the Syntax tab is executed). This output might include R or Python error messages or warnings that are produced when the R or Python script is executed. The output can be used, primarily, to debug the script. The Console Output tab also contains the script from the R Syntax or Python Syntax field.

Every time the Extension Transform script is executed, the content of the Console Output tab is overwritten with the output received from the R console or Python for Spark. The output cannot be edited.

Extension Import node

With the Extension Import node, you can run R or Python for Spark scripts to import data.

- [Extension Import node - Syntax tab](#)
- [Extension Import node - Console Output tab](#)
- [Filtering or renaming fields](#)
- [Viewing and setting information about types](#)

Extension Import node - Syntax tab

Select your type of syntax – R or Python for Spark. Then enter or paste your custom script for importing data. When your syntax is ready, you can click Run to execute the Extension Import node.

R example

```
# import R demo data cars to modeler
modelerData <- cars

# write the data model that matches the data
var1<-c(fieldName="speed",fieldLabel="",fieldStorage="integer",fieldMeasure="",fieldFormat="", fieldRole="")
var2<-c(fieldName="dist",fieldLabel="",fieldStorage="integer",fieldMeasure="",fieldFormat="", fieldRole="")
modelerDataModel<-data.frame(var1, var2)
```

Python for Spark example

```
import spss.pyspark.runtime
from pyspark.sql import SQLContext
from pyspark.sql.types import *

ctx = spss.pyspark.runtime.getContext()
if ctx.isComputeDataModelOnly():
    _schema = StructType([StructField("Age", LongType(), nullable=True), \
                          StructField("Sex", StringType(), nullable=True), \
                          StructField("BP", StringType(), nullable=True), \
                          StructField("Cholesterol", StringType(), nullable=True), \
                          StructField("Na", DoubleType(), nullable=True), \
                          StructField("K", DoubleType(), nullable=True), \
                          StructField("Drug", StringType(), nullable=True)])
    ctx.setSparkOutputSchema(_schema)
else:
    sqlContext = ctx.getSparkSQLContext()
    # the demo data is in modeler installation path
    df = sqlContext.read.option("inferSchema", "true").option("header", "true").csv("/opt/IBM/SPSS/ModelerServer/Cloud/demos/DRUG1n")
    ctx.setSparkOutputData(df)
    df.show()
    # print (df.dtypes[:])
```

Extension Import node - Console Output tab

The Console Output tab contains any output that is received when the R script or Python for Spark script on the Syntax tab runs (for example, if using an R script, it shows output received from the R console when the R script in the R Syntax field on the Syntax tab is executed). This output might include R or Python error messages or warnings that are produced when the R or Python script is executed. The output can be used, primarily, to debug the script. The Console Output tab also contains the script from the R Syntax or Python Syntax field.

Every time the Extension Import script is executed, the content of the Console Output tab is overwritten with the output received from the R console or Python for Spark. The output cannot be edited.

Filtering or renaming fields

You can rename or exclude fields at any point in a stream. For example, as a medical researcher, you may not be concerned about the potassium level (field-level data) of patients (record-level data); therefore, you can filter out the `K` (potassium) field. This can be done using a separate Filter node or using the Filter tab on a source or output node. The functionality is the same regardless of which node it is accessed from.

- From source nodes, such as Variable File, Fixed File, Statistics File, XML, or Extension Import, you can rename or filter fields as the data are read into IBM® SPSS® Modeler.
- Using a Filter node, you can rename or filter fields at any point in the stream.

- From Statistics Export, Statistics Transform, Statistics Model, and Statistics Output nodes, you can filter or rename fields to conform to IBM SPSS Statistics naming standards. See the topic [Renaming or Filtering Fields for](#) more information.
 - You can use the Filter tab in any of the above nodes to define or edit multiple response sets. See the topic [Editing Multiple Response Sets](#) for more information.
 - Finally, you can use a Filter node to map fields from one source node to another.
-

Viewing and setting information about types

From various source nodes as well as the Type node, you can specify field metadata and properties that are invaluable to modeling and other work in IBM® SPSS® Modeler. These properties include:

- Specifying a usage type, such as range, set, ordered set, or flag, for each field in your dataset.
- Setting options for handling missing values and system nulls.
- Setting the role of a field for modeling purposes.
- Specifying values for a field as well as options used to automatically read values from the dataset.
- Specifying field and value labels.

Select help specific to your situation from the list below.

Extensions

Extensions are custom components that extend the capabilities of IBM® SPSS® Modeler. Extensions are packaged in extension bundles (.mpe files) and are installed to IBM SPSS Modeler. Extensions can be created by any user and shared with other users by sharing the associated extension bundle.

The following utilities are provided for working with extensions:

- The [Extension Hub](#), which is accessed from Extensions > Extension Hub, is an interface for searching for, downloading, and installing extensions from the IBM SPSS Predictive Analytics collection on GitHub. From the Extension Hub dialog, you can also view details of the extensions that are installed on your computer, get updates for installed extensions, and remove extensions.
 - You can install an extension bundle that is stored on your local computer from Extensions > Install Local Extension Bundle.
 - You can use the [Custom Dialog Builder for Extensions](#) to create an extension that includes a user interface, which is referred to as a custom node dialog. Custom node dialogs generate R script or Python for Spark script that carries out the tasks that are associated with the extension. You design the generated script as part of designing the custom dialog.
- [Extension Hub](#)**
• [Installing local extension bundles](#)
• [Creating and managing custom nodes](#)
-

Extension Hub

From the Extension Hub dialog, you can do the following tasks:

- Explore extensions that are available from the IBM® SPSS® Predictive Analytics collection on GitHub. You can select extensions to install now or you can download selected extensions and install them later.
- Get updated versions of extensions that are already installed on your computer.
- View details about the extensions that are already installed on your computer.
- Remove extensions that are installed on your computer.

To download or remove extensions:

1. From the menus, choose: Extensions > Extension Hub
2. Select the extensions that you want to download or remove and click OK. All selections that are made on the Explore and Installed tabs are processed when you click OK.

By default, the extensions that are selected for download are downloaded and installed on your computer. From the Settings tab, you can choose to download the selected extensions to a specified location without installing them. You can then install them later by choosing Extensions > Install Local Extension Bundle. For Windows, you can install an extension by double-clicking the extension bundle file.

Important:

- For Windows 7 and later, installing an updated version of an existing extension bundle might require running IBM SPSS Modeler with administrator privileges. You can start IBM SPSS Modeler with administrator privileges by right-clicking the icon for IBM SPSS Modeler and

choosing Run as administrator. In particular, if you receive an error message that states that one or more extension bundles could not be installed, then try running with administrator privileges.

- If connecting to the Internet via a proxy, you may receive an error like "Some features are not available because no Internet connection was detected" when you try to open the Extension Hub via the Extensions > Extension Hub menu option. To resolve this, you add the following parameters to the #
JVM options in the jvm.cfg file (located in the config directory of your SPSS Modeler installation). Save the file and restart SPSS Modeler.

```
options, "-DproxyHost=proxyIP"  
options, " -DproxyPort=proxyPort"
```

Note: The license that you agree to when you install an extension can be viewed at any later time by clicking More info... for the extension on the Installed tab.

- [Explore tab \(Extension Hub\)](#)
- [Installed tab \(Extension Hub\)](#)
- [Settings \(Extension Hub\)](#)
- [Extension Details](#)

Explore tab (Extension Hub)

The Explore tab displays all of the extensions that are available from the IBM® SPSS® Predictive Analytics collection on GitHub (<https://ibmpredictiveanalytics.github.io/>). From the Explore tab, you can select new extensions to download and install, and you can select updates for extensions that are already installed on your computer. The Explore tab requires an internet connection.

- For each extension, the number of the latest version and the associated date of that version are displayed. A brief summary of the extension is also provided. For extensions that are already installed on your computer, the installed version number is also displayed.
- You can view detailed information about an extension by clicking More info. When an update is available, More info displays information about the update.
- You can view the prerequisites for running an extension, such as whether the IBM SPSS Modeler - Integration Plug-in for R is required, by clicking Prerequisites. When an update is available, Prerequisites displays information about the update.

Refine by

You can refine the set of extensions that are displayed. You can refine by general categories of extensions, the language in which the extension is implemented, the type of organization that provided the extension, or the state of the extension. For each group, such as Category, you can select multiple items by which to refine the displayed list of extensions. You can also refine by search terms. Searches are not case-sensitive, and the asterisk (*) is treated as any other character and does not indicate a wildcard search.

- To refine the displayed list of extensions, click Apply. Pressing the enter key when the cursor is in the Search box has the same effect as clicking Apply.
- To reset the list to display all available extensions, delete any text in the Search box, deselect all items, and click Apply.
- [How to get integration plug-ins](#)

How to get integration plug-ins

To get the IBM® SPSS® Modeler - Integration Plug-in for R:

Install IBM SPSS Modeler - Essentials for R, available from https://github.com/IBMPredictiveAnalytics/R_Essentials_Modeler/releases/ or the IBM SPSS Statistics community at <https://www.ibm.com/products/spss-statistics/support>. IBM SPSS Modeler - Essentials for R includes the IBM SPSS Modeler - Integration Plug-in for R. Essentials for R does not include the R programming language. Before installing IBM SPSS Modeler - Essentials for R you will need to install R version 4.0 if it is not already installed. It is available from <https://cran.r-project.org/>. It is recommended to download and install R 4.0.x.

Note: If you are installing Essentials for R on a computer that does not have internet access and you plan to use the R scripts that are included with Essentials for R, then you must obtain any R packages that are required by those scripts and manually install them in R. To determine which R packages are required for a specific R script, open the Extension Hub dialog (Extensions > Extension Hub), click the Installed tab, and then click More info for the desired extension. The required R packages are listed on the Extension Details dialog. R packages can be obtained from any of the R CRAN mirror sites, which are accessed from <http://www.r-project.org/>. Be sure to obtain the versions of the packages that match your R version. The version-specific packages are available from links on the "Contributed Packages" page of the CRAN mirror site.

Installed tab (Extension Hub)

The Installed tab displays all of the extensions that are installed on your computer. From the Installed tab, you can select updates for installed extensions that are available from the IBM® SPSS® Predictive Analytics collection on GitHub and you can remove extensions. To get updates for installed extensions, you must have an internet connection.

- For each extension, the installed version number is displayed. When an internet connection is available, the number of the latest version and the associated date of that version are displayed. A brief summary of the extension is also provided.
- You can view detailed information about an extension by clicking More info. When an update is available, More info displays information about the update.
- You can view the prerequisites for running an extension, such as whether the IBM SPSS Modeler - Integration Plug-in for R is required, by clicking Prerequisites. When an update is available, Prerequisites displays information about the update.

Refine by

You can refine the set of extensions that are displayed. You can refine by general categories of extensions, the language in which the extension is implemented, the type of organization that provided the extension, or the state of the extension. For each group, such as Category, you can select multiple items by which to refine the displayed list of extensions. You can also refine by search terms. Searches are not case-sensitive, and the asterisk (*) is treated as any other character and does not indicate a wildcard search.

- To refine the displayed list of extensions, click Apply. Pressing the enter key when the cursor is in the Search box has the same effect as clicking Apply.
- To reset the list to display all available extensions, delete any text in the Search box, deselect all items, and click Apply.

Private extensions

Private extensions are extensions that are installed on your computer but are not available from the IBM SPSS Predictive Analytics collection on GitHub. The features for refining the set of displayed extensions and for viewing the prerequisites to run an extension are not available for private extensions.

Note: When using the Extension Hub without an internet connection, some of the features of the Installed tab might not be available.

Related information

- [Extension Details](#)

Settings (Extension Hub)

The Settings tab specifies whether extensions that are selected for download are downloaded and then installed or downloaded but not installed. This setting applies to new extensions and updates to existing extensions. You might choose to download extensions without installing them if you are downloading extensions to distribute to other users within your organization. You might also choose to download, but not install, extensions if you don't have the prerequisites for running the extensions but plan to get the prerequisites.

If you choose to download extensions without installing them, you can install them later by choosing Extensions...>Install Local Extension Bundle. For Windows, you can install an extension by double-clicking the extension bundle file.

Extension Details

The Extension Details dialog box displays the information that was provided by the author of the extension. In addition to required information, such as Summary, and Version, the author might have included URLs to locations of relevance, such as the author's home page. If the extension was downloaded from the Extension Hub, then it includes a license that can be viewed by clicking View license.

Custom Nodes. The Custom Nodes table lists the custom node dialogs that are included in the extension.

Note: Installing an extension that contains a custom node dialog might require a restart of IBM® SPSS® Modeler to see the entry for the node dialog in the Custom Nodes table.

Dependencies. The Dependencies group lists add-ons that are required to run the components included in the extension.

- Integration Plug-In for R. The components for an extension may require the Integration Plug-in for R.
- R packages. Lists any R packages that are required by the extension. See the topic [Required R packages](#) for more information.

How To Access the Details for an Installed Extension

1. From the menus, choose:
Extensions...>Extension Hub

2. Click the Installed tab on the Extension Hub dialog.
3. Click More info for the desired extension.

Related information

- [Creating and editing extension bundles](#)
 - [Installing local extension bundles](#)
-

Installing local extension bundles

To install an extension bundle that is stored on your local computer:

1. From the menus choose:
Extensions > Install Local Extension Bundle...
2. Select the extension bundle. Extension bundles have a file type of mpe.

Important: For users of Windows 7 and later versions of Windows, installing an updated version of an existing extension bundle might require running IBM® SPSS® Modeler with administrator privileges. You can start IBM SPSS Modeler with administrator privileges by right-clicking the icon for IBM SPSS Modeler and choosing Run as administrator. In particular, if you receive an error message that states that one or more extension bundles could not be installed, then try running with administrator privileges.

- [Installation locations for extensions](#)
- [Required R packages](#)

Related information

- [Extension Details](#)
-

Installation locations for extensions

By default, extensions are installed to a general user-writable location for your operating system.

You can override the default location by defining a path with the IBM_SPSS_MODELER_EXTENSION_PATH environment variable. The specified location must exist on the target computer. After you set IBM_SPSS_MODELER_EXTENSION_PATH, you must restart IBM® SPSS® Modeler for the changes to take effect.

To create an environment variable on Windows, from the Control Panel:

Windows 7

1. Select User Accounts.
2. Select Change my environment variables.
3. Click New, enter the name of the environment variable (for instance, IBM_SPSS_MODELER_EXTENSION_PATH) in the Variable name field and enter the path or paths in the Variable value field.

Windows 8 or Later

1. Select System.
2. Select the Advanced tab and click Environment Variables. The Advanced tab is accessed from Advanced system settings.
3. In the User variables section, click New, enter the name of the environment variable (for instance, IBM_SPSS_MODELER_EXTENSION_PATH) in the Variable name field and enter the path or paths in the Variable value field.

Important: For users of Windows 7 and later versions of Windows, installing an updated version of an existing extension bundle might require running IBM SPSS Modeler with administrator privileges. You can start IBM SPSS Modeler with administrator privileges by right-clicking the icon for IBM SPSS Modeler and choosing Run as administrator. In particular, if you receive an error message that states that one or more extension bundles could not be installed, then try running with administrator privileges.

Running streams in batch mode that contain extensions

To use Custom Dialog Builder extensions with Modeler Batch:

1. In Modeler Client, use Custom Dialog Builder to create and install your custom node. The extension node files will be in the following directory, by default:
 - Windows: C:\ProgramData\IBM\SPSS\Modeler\<version>\CDB\
 - Mac: /Users/yourname/Library/Application Support/IBM/SPSS\Modeler/<version>/CDB/
2. Copy the CDB directory to the Modeler Batch machine.
3. Export the environment variable **IBM_SPSS_MODELER_EXTENSION_PATH** to the new directory. You can now run the stream in batch mode.

Required R packages

If you do not have internet access, you will need to obtain any required R packages for a particular extension, that are not found on your computer, from someone who does. You can view the list of required R packages from the Extension Details dialog box, once the extension is installed. See the topic [Extension Details](#) for more information. Packages can be downloaded from <http://www.r-project.org/> and then installed from within R. For details, see the *R Installation and Administration* guide, distributed with R.

Note: For UNIX (including Linux) users, packages are downloaded in source form and then compiled. This requires that you have the appropriate tools installed on your machine. See the *R Installation and Administration* guide for details. In particular, Debian users should install the `r-base-dev` package from `apt-get install r-base-dev`.

Creating and managing custom nodes

The Custom Dialog Builder for Extensions creates nodes to use inside SPSS® Modeler streams.

Using the Custom Dialog Builder for Extensions you can:

- Create a custom node dialog for executing a node that is implemented in R, or in Apache Spark (via Python). See [Building the script template](#) for more information.
- Open a file containing the specification for a custom node dialog--perhaps created by another user--and add the dialog to your installation of IBM® SPSS Modeler, optionally making your own modifications.
- Save the specification for a custom node dialog so that other users can add it to their installations of IBM SPSS Modeler.
- Create custom nodes and write Python for Spark scripts to read data from wherever your data source is, and write data out to any data format supported by Apache Spark. See [Importing and exporting data using Python for Spark](#) for more information.
- Create custom nodes and write R scripts to read data from wherever your data source is, and write data out to any data format supported by R. See [Importing and exporting data using R](#) for more information.

In the Custom Dialog Builder for Extensions, you create or modify custom node dialogs within extensions. When you open the Custom Dialog Builder for Extensions, a new extension that contains an empty custom node dialog is created. When you save or install custom node dialogs from the Custom Dialog Builder for Extensions, they are saved or installed as part of an extension.

Note:

- You cannot create your own version of a node dialog for a standard IBM SPSS Modeler node.
- Scripting is not supported for nodes that are created with Custom Dialog Builder, including Custom Dialog Builder R nodes and Custom Dialog Builder Python nodes.

How to start the Custom Dialog Builder for Extensions

From the menus, choose **Extensions > Custom Node Dialog Builder**

Note:

- Python nodes depend on the Spark environment.
- Python scripts must use the Spark API because data will be presented in the form of a Spark DataFrame.
- Old nodes created in version 17.1 will still only run against IBM SPSS Analytic Server (the data originates from an IBM SPSS Analytic Server source node and has not been extracted to IBM SPSS Modeler Server). New Python and Custom Dialog Builder nodes created in version 18.0 or later can run against IBM SPSS Modeler Server.
- When installing Python, make sure all users have permission to access the Python installation.
- If you want to use the Machine Learning Library (MLlib), you must install a version of Python that includes NumPy. Then you must configure the IBM SPSS Modeler Server (or the local server in IBM SPSS Modeler Client) to use your Python installation. For details, see [Scripting with Python for Spark](#).
- [Custom Dialog Builder layout](#)
- [Building a custom node dialog](#)
- [Dialog Properties](#)
- [Laying out controls on the dialog canvas](#)

- [Building the script template](#)
 - [Previewing a custom node dialog](#)
 - [Control types](#)
 - [Extension Properties](#)
 - [Managing custom node dialogs](#)
 - [Creating Localized Versions of Custom Node Dialogs](#)
 - [Importing and exporting data using Python for Spark](#)
 - [Importing and exporting data using R](#)
-

Custom Dialog Builder layout

Dialog Canvas

The dialog canvas is the area of the Custom Dialog Builder where you design the layout of your node dialog.

Properties Pane

The properties pane is the area of the Custom Dialog Builder where you specify properties of the controls that make up the node dialog as well as properties of the dialog itself, such as the node type.

Tools Palette

The tools palette provides the set of controls that can be included in a custom node dialog. You can show or hide the Tools Palette by choosing Tools Palette from the View menu.

Script Template

The Script Template specifies the R script or Python for Spark script that is generated by the custom node dialog. You can move the Script Template pane to a separate window by clicking Move to New Window. To move a separate Script Template window back into the Custom Dialog Builder, click Restore to Main Window.

Related information

- [Creating and managing custom nodes](#)
 - [Building a custom node dialog](#)
 - [Dialog Properties](#)
 - [Laying out controls on the dialog canvas](#)
 - [Building the script template](#)
 - [Previewing a custom node dialog](#)
 - [Managing custom node dialogs](#)
 - [Control types](#)
 - [Creating Localized Versions of Custom Node Dialogs](#)
-

Building a custom node dialog

The basic steps involved in building a custom node dialog are:

1. Specify the properties of the node dialog itself, such as the title that appears when the node dialog is launched and the location of the new node within the IBM® SPSS® Modeler palettes. See the topic [Dialog Properties](#) for more information.
2. Specify the controls, such as field choosers and check boxes, that make up the node dialog and any sub-dialogs. See the topic [Control types](#) for more information.
3. Create the script template that specifies the R code or Python for Spark code that is generated by the node dialog. See the topic [Building the script template](#) for more information.
4. Specify properties of the extension that contains your node dialog. See the topic [Extension Properties](#) for more information.
5. Install the extension that contains the node dialog to IBM SPSS Modeler and/or save the extension to an extension bundle (.mpe) file. See the topic [Managing custom node dialogs](#) for more information.

You can preview your node dialog as you're building it. See the topic [Previewing a custom node dialog](#) for more information.

Related information

- [Creating and managing custom nodes](#)
 - [Custom Dialog Builder layout](#)
 - [Dialog Properties](#)
 - [Laying out controls on the dialog canvas](#)
 - [Building the script template](#)
 - [Previewing a custom node dialog](#)
 - [Managing custom node dialogs](#)
 - [Control types](#)
 - [Creating Localized Versions of Custom Node Dialogs](#)
-

Dialog Properties

The Custom Dialog Builder window shows the properties for the node dialog and for the selected user interface control. To view and set Dialog Properties, click on the canvas in an area outside of any controls. With no controls on the canvas, Dialog Properties are always visible.

Dialog Name. The Dialog Name property is required and specifies a unique name to associate with the node dialog. To minimize the possibility of name conflicts, you may want to prefix the name with an identifier for your organization, such as a URL.

Title. The Title property specifies the text to be displayed in the title bar of the node dialog box.

Help File. The Help File property is optional and specifies the path to a help file for the node dialog. This is the file that will be launched when the user clicks the Help button on the dialog. Help files must be in HTML format. A copy of the specified help file is included with the specifications for the node dialog when the node dialog is installed or saved. The Help button on the run-time dialog is hidden if there is no associated help file.

- Localized versions of the help file that exist in the same directory as the help file are automatically added to the node dialog when you add the help file. Localized versions of the help file are named <Help File>_<language identifier>.htm. For more information, see the topic [Creating Localized Versions of Custom Node Dialogs](#).
- Supporting files, such as image files and style sheets, can be added to the node dialog by first saving the node dialog. You then manually add the supporting files to the node dialog file (.cfe). For information about accessing and manually modifying custom node dialog files, see the section that is titled "To localize dialog strings" in the topic [Creating Localized Versions of Custom Node Dialogs](#).

Script Type. Specifies the type of script that can be used to build the Script Template. In IBM® SPSS® Modeler, R scripting or Python for Spark scripting can be used.

Score from the Model. Specifies whether the model that is built using the model building script is to be used for scoring.

Node Type. Specifies the type of node that will be created when you install your node dialog.

Palette. Specifies the palette to which the newly created node will be added when you install your node dialog.

Node Icon. Click the ellipsis (...) button to select an image to be used as the node icon for the newly created node. The image that you choose must be a .gif file.

Related information

- [Creating and managing custom nodes](#)
 - [Custom Dialog Builder layout](#)
 - [Building a custom node dialog](#)
 - [Laying out controls on the dialog canvas](#)
 - [Building the script template](#)
 - [Previewing a custom node dialog](#)
 - [Managing custom node dialogs](#)
 - [Control types](#)
 - [Creating Localized Versions of Custom Node Dialogs](#)
 - [Extension Properties](#)
-

Laying out controls on the dialog canvas

You add controls to a custom node dialog by dragging them from the tools palette onto the dialog canvas. To ensure consistency with built-in node dialogs, the dialog canvas is divided into four functional columns in which you can place controls.

- The first (leftmost) column is primarily intended for a Field Chooser control.
- Sub-dialog buttons must be in the rightmost column (for example, the third column if only three columns are used) and no other controls can be in the same column as Sub-dialog buttons. In that regard, the fourth column can contain only Sub-dialog buttons.

Although not shown on the dialog canvas, when the node dialog is installed into IBM® SPSS® Modeler, the appropriate buttons are added to the dialog (for example: OK, Cancel, Apply, Reset and, if appropriate, Help and Run). The presence and locations of these buttons is automatic. However, the Help button is hidden if there is no help file associated with the node dialog (as specified by the Help File property in Dialog Properties).

You can change the vertical order of the controls within a column by dragging them up or down, but the exact position of the controls is determined automatically for you. At run-time, controls resize in appropriate ways when the dialog itself is resized. Controls such as field choosers automatically expand to fill the available space below them.

Related information

- [Creating and managing custom nodes](#)
 - [Custom Dialog Builder layout](#)
 - [Building a custom node dialog](#)
 - [Dialog Properties](#)
 - [Building the script template](#)
 - [Previewing a custom node dialog](#)
 - [Managing custom node dialogs](#)
 - [Control types](#)
 - [Creating Localized Versions of Custom Node Dialogs](#)
-

Building the script template

The script template specifies the R script or Python for Spark script that the custom node dialog will generate. A single custom node dialog can be used to specify one or more operations which will run in sequence.

The script template might consist of *static text*. Static text is different to the static text control; it is R code or Python for Spark code that is always generated when the node runs. For example, command names and subcommand specifications that don't depend on user input are static text. The script template might also consist of control identifiers that are replaced at run-time with the values of the associated custom node dialog controls. For example, the set of fields specified in a field chooser is represented with the control identifier for the field chooser control.

To build the Script Template

1. For static text that does not depend on user-specified values, enter the R script or Python for Spark script as you would in, for example, the R model building syntax field of the R Build node.
2. Add control identifiers of the form `%%Identifier%%` at the locations where you want to insert R script or Python for Spark script generated by controls, where `Identifier` is the value of the Identifier property for the control.
 - You can insert a control identifier by selecting a row in the table of identifiers, right-clicking and selecting Add to script template.
 - You can also insert a control identifier by right-clicking a control on the canvas and selecting Add to script template.
 - You can also select from a list of available control identifiers by pressing **Ctrl+Spacebar**. The list contains the control identifiers followed by the items available with the script auto-completion feature.

If you manually enter identifiers, retain any spaces, since all spaces in identifiers are significant.

At run-time, and for all controls other than check boxes, check box groups, and the static text control, each identifier is replaced with the current value of the Script property of the associated control. If the control is empty at run-time, it does not generate any script. For check boxes and check box groups, the identifier is replaced by the current value of the Checked R Script or Unchecked R Script property of the associated control, depending on the current state of the control--checked or unchecked. See the topic [Control types](#) for more information.

Example: Including run-time values in an R script template

In this example, the custom node dialog will generate and run R script to build and score a linear regression model, using a call to the R `lm` function with the signature shown here.

```
lm(formula, data)
```

- `formula` specifies an expression, such as `Na~Age`, where `Na` is the target field of the model, and the input field of the model is `Age`.
- `data` is a data frame containing the values of the fields that are specified in the formula.

Consider a custom node dialog with a single field chooser control that allows the user to choose the input field of the linear model. The script template to generate and run the R script that builds the model is entered on the Script tab, and might look like this:

```
modelerModel <- lm(Na~%%input%%, data=modelerData)
```

- `%%input%%` is the value of the Identifier property for the field chooser control. At run-time it will be replaced by the current value of the Script property of the control.

- Defining the Script property of the field chooser control to be `%%ThisValue%%` specifies that at run-time the current value of the property will be the value of the control, which is the field that is chosen from the field chooser.

Suppose the user of the custom node dialog selects the Age field as the input field of the model. The following R script is then generated by the node dialog:

```
modelerModel <- lm(Na~Age,data=modelerData)
```

The script template to generate and run the R script that scores the model is entered on the Score Script tab, and might look like this:

```
result <- predict(modelerModel,newdata=modelerData)
var1 <-c(fieldName="predicted", fieldLabel="",fieldStorage="real",fieldMeasure="",fieldFormat="",
fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)
```

This R script does not depend on any user-specified values, only on the model that is built using the model building R script. Therefore, the model scoring R script is entered as it would be in the R model scoring syntax field of the R Build node.

Related information

- [Creating and managing custom nodes](#)
- [Custom Dialog Builder layout](#)
- [Building a custom node dialog](#)
- [Dialog Properties](#)
- [Laying out controls on the dialog canvas](#)
- [Previewing a custom node dialog](#)
- [Managing custom node dialogs](#)
- [Control types](#)
- [Creating Localized Versions of Custom Node Dialogs](#)

Previewing a custom node dialog

You can preview the node dialog that is currently open in the Custom Dialog Builder. The dialog appears and functions as it would when run from a node within IBM® SPSS® Modeler.

- Field choosers are populated with dummy fields.
- The OK button closes the preview.
- If a help file is specified, the Help button is enabled and will open the specified file. If no help file is specified, the help button is disabled when previewing, and hidden when the actual dialog is run.

To preview a custom node dialog, from the menus in the Custom Dialog Builder, choose File> Preview Dialog.

- [A help file has not been specified for this node](#)

Related information

- [Creating and managing custom nodes](#)
- [Custom Dialog Builder layout](#)
- [Building a custom node dialog](#)
- [Dialog Properties](#)
- [Laying out controls on the dialog canvas](#)
- [Building the script template](#)
- [Managing custom node dialogs](#)
- [Control types](#)
- [Creating Localized Versions of Custom Node Dialogs](#)

Control types

The tools palette provides all of the standard controls that might be needed in a custom node dialog.

- Field Chooser: A list of all the fields from the active dataset. See the topic [Field Chooser](#) for more information.
- Check Box: A single check box. See the topic [Check Box](#) for more information.
- Combo Box: A combo box for creating drop-down lists. See the topic [Combo Box](#) for more information.
- List Box: A list box for creating single selection or multiple selection lists. See the topic [Combo Box](#) for more information.
- Text control: A text box that accepts arbitrary text as input. See the topic [Text control](#) for more information.

- Number control: A text box that is restricted to numeric values as input. See the topic [Number Control](#) for more information.
 - Date control: A spinner control for specifying date/time values, which include dates, times, and datetimes. See the topic [Date control](#) for more information.
 - Secured Text: A text box that masks user entry with asterisks. See the topic [Secured Text](#) for more information.
 - Static Text control: A control for displaying static text. See the topic [Static Text Control](#) for more information.
 - Color Picker: A control for specifying a color and generating the associated RGB value. See the topic [Color Picker](#) for more information.
 - Table control: A table with a fixed number of columns and a variable number of rows that are added at run time. See the topic [Table Control](#) for more information.
 - Item Group: A container for grouping a set of controls, such as a set of check boxes. See the topic [Item Group](#) for more information.
 - Radio Group: A group of radio buttons. See the topic [Radio Group](#) for more information.
 - Check Box Group: A container for a set of controls that are enabled or disabled as a group, by a single check box. See the topic [Check Box Group](#) for more information.
 - File Browser: A control for browsing the file system to open or save a file. See the topic [File Browser](#) for more information.
 - Tab: A single tab. See the topic [Tab](#) for more information.
 - Sub-dialog Button: A button for launching a sub-dialog. See the topic [Sub-dialog button](#) for more information.
- [Field Chooser](#)**
• [Filtering Field Lists](#)
• [Check Box](#)
• [Combo Box](#)
• [List Box](#)
• [Text control](#)
• [Number Control](#)
• [Date control](#)
• [Secured Text](#)
• [Static Text Control](#)
• [Color Picker](#)
• [Table Control](#)
• [Item Group](#)
• [Radio Group](#)
• [Check Box Group](#)
• [File Browser](#)
• [Tab](#)
• [Sub-dialog button](#)
• [Specifying an Enabling Rule for a Control](#)
-

Field Chooser

The Field Chooser control displays the list of fields that are available to the end user of the node dialog. You can display all fields from the active dataset (the default) or you can filter the list based on type and measurement level--for instance, numeric fields that have a measurement level of scale. You can also specify any other Field Chooser as the source of fields for the current Field Chooser. The Field Chooser control has the following properties:

Identifier. The unique identifier for the control.

Title. An optional title that appears above the control. For multi-line titles, use \n to specify line breaks.

Title Position. Specifies the position of the title relative to the control. Values are Top and Left where Top is the default. This property only applies when the chooser type is set to select a single field.

ToolTip. Optional ToolTip text that appears when the user hovers over the control. The specified text only appears when hovering over the title area of the control. Hovering over one of the listed fields displays the field name and label.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

Chooser Type. Specifies whether the Field Chooser in the custom node dialog can be used to select a single field or multiple fields from the field list.

Separator Type. Specifies the delimiter between the selected fields in the generated script. The allowed separators are a blank, a comma, and a plus sign (+). You can also enter an arbitrary single character to be used as the separator.

Minimum Fields. The minimum number of fields that must be specified for the control, if any.

Maximum Fields. The maximum number of fields that can be specified for the control, if any.

Required for Execution. Specifies whether a value is required in this control in order for execution to proceed. If True is specified, the user of the node dialog must specify a value for the control otherwise clicking the OK button will generate an error. If False is specified, the absence of a

value in this control has no effect on the state of the OK button.

Variable Filter. Allows you to filter the set of fields that are displayed in the control. You can filter on field type and measurement level, and you can specify that multiple response sets are included in the field list. Click the ellipsis (...) button to open the Filter dialog. You can also open the Filter dialog by double-clicking the Field Chooser control on the canvas. See the topic [Filtering Field Lists](#) for more information.

Field Source. Specifies that another Field Chooser is the source of fields for the current Field Chooser. When the Field Source property is not set, the source of fields is the active dataset. Click the ellipsis (...) button to open the [Field Source](#) dialog box and specify the field source.

Script. Specifies the script that is generated and run by this control at run-time and can be inserted in the script template.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.
- The value `%%ThisValue%%` specifies the run-time value of the control, which is the list of fields. This is the default.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

- [Specifying the Field Source for a Field Chooser](#)

Related information

- [Control types](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Combo Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Text control](#)
- [Number Control](#)
- [Static Text Control](#)
- [Item Group](#)
- [Radio Group](#)
- [Defining radio buttons](#)
- [Check Box Group](#)
- [File Browser](#)
- [File type filter](#)
- [Tab](#)
- [Sub-dialog button](#)
- [Dialog properties for a sub-dialog](#)

Specifying the Field Source for a Field Chooser

The Field Source dialog box specifies the source of the fields that are displayed in the Field Chooser. The source can be any other Field Chooser. You can choose to display the fields that are in the selected control or the fields, from the active dataset, that are not in the selected control.

Filtering Field Lists

The Filter dialog box, associated with field chooser controls, allows you to filter the types of fields from the active dataset that can appear in the lists. You can also specify whether multiple response sets associated with the active dataset are included. Numeric fields include all numeric formats except date and time formats.

Related information

- [Control types](#)
- [Field Chooser](#)
- [Check Box](#)
- [Combo Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Text control](#)
- [Number Control](#)
- [Static Text Control](#)

- [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Sub-dialog button](#)
 - [Dialog properties for a sub-dialog](#)
-

Check Box

The Check Box control is a simple check box that can generate and run different R scripts or Python for Spark scripts for the checked versus the unchecked state. The Check Box control has the following properties:

Identifier. The unique identifier for the control.

Title. An optional title that appears above the control. For multi-line titles, use \n to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

Default Value. The default state of the check box--checked or unchecked.

Checked/Unchecked Script. Specifies the R script or Python for Spark script that is generated and run when the control is checked and when it is unchecked. To include the script in the script template, use the value of the Identifier property. The generated script, whether from the Checked Script or Unchecked Script property, will be inserted at the specified positions of the identifier. For example, if the identifier is *checkbox1*, then at run time, instances of %%checkbox1%% in the script template will be replaced by the value of the Checked Script property when the box is checked and the Unchecked Script property when the box is unchecked.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Related information

- [Control types](#)
 - [Field Chooser](#)
 - [Filtering Field Lists](#)
 - [Combo Box](#)
 - [Specifying list items for combo boxes and list boxes](#)
 - [Text control](#)
 - [Number Control](#)
 - [Static Text Control](#)
 - [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Sub-dialog button](#)
 - [Dialog properties for a sub-dialog](#)
-

Combo Box

The Combo Box control allows you to create a drop-down list that can generate and run R script or Python for Spark script specific to the selected list item. It is limited to single selection. The Combo Box control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title that appears above the control. For multi-line titles, use \n to specify line breaks.

Title Position. Specifies the position of the title relative to the control. Values are Top and Left where Top is the default.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

List Items. Click the ellipsis (...) button to open the [List Item Properties](#) dialog box, which allows you to specify the list items of the control. You can also open the List Item Properties dialog by double-clicking the Combo Box control on the canvas.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

Editable. Specifies whether the Combo Box control is editable. When the control is editable, a custom value can be entered at run time.

Script. Specifies the R script or Python for Spark script that is generated by this control at run time and can be inserted in the script template.

- The value %%**ThisValue**%% specifies the run time value of the control and is the default. If the list items are manually defined, the run time value is the value of the Script property for the selected list item. If the list items are based on a target list control, the run time value is the value of the selected list item. For multiple selection list box controls, the run time value is a blank-separated list of the selected items. See the topic [Specifying list items for combo boxes and list boxes](#) for more information.
- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.

Quote Handling. Specifies handling of quotation marks in the run time value of %%**ThisValue**%% when the Script property contains %%**ThisValue**%% as part of a quoted string. In this context, a quoted string is a string that is enclosed in single quotation marks or double quotation marks. Quote handling applies only to quotation marks that are the same type as the quotation marks that enclose %%**ThisValue**%%. The following types of quote handling are available.

Python

Quotation marks in the run time value of %%ThisValue%% that match the enclosing quotation marks are escaped with the backslash character (\). For example, if the Script property is '%%%ThisValue%%%' and the run time value of the combo box is **Combo box's value**, then the generated script is '**Combo box\'s value**'. Note that quote handling is not done when %%ThisValue%% is enclosed in triple quotation marks.

R

Quotation marks in the run time value of %%ThisValue%% that match the enclosing quotation marks are escaped with the backslash character (\). For example, if the Script property is '%%%ThisValue%%%' and the run time value of the combo box is **Combo box's value**, then the generated script is '**Combo box\'s value**'.

None

Quotation marks in the run time value of %%ThisValue%% that match the enclosing quotation marks are retained with no modification.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

- [Specifying list items for combo boxes and list boxes](#)

Related information

- [Control types](#)
- [Field Chooser](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Text control](#)
- [Number Control](#)
- [Static Text Control](#)
- [Item Group](#)
- [Radio Group](#)
- [Defining radio buttons](#)
- [Check Box Group](#)
- [File Browser](#)
- [File type filter](#)
- [Tab](#)
- [Sub-dialog button](#)
- [Dialog properties for a sub-dialog](#)

Specifying list items for combo boxes and list boxes

The List Item Properties dialog box allows you to specify the list items of a combo box or list box control.

Manually defined values. Allows you to explicitly specify each of the list items.

- Identifier. A unique identifier for the list item.
- Name. The name that appears in the list for this item. The name is a required field.
- Default. For a combo box, specifies whether the list item is the default item displayed in the combo box. For a list box, specifies whether the list item is selected by default.
- Script. Specifies the R script or Python for Spark script that is generated when the list item is selected.
- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.

Note: You can add a new list item in the blank line at the bottom of the existing list. Entering any of the properties other than the identifier will generate a unique identifier, which you can keep or modify. You can delete a list item by clicking on the *Identifier* cell for the item and pressing delete.

Related information

- [Control types](#)
 - [Field Chooser](#)
 - [Filtering Field Lists](#)
 - [Check Box](#)
 - [Combo Box](#)
 - [Text control](#)
 - [Number Control](#)
 - [Static Text Control](#)
 - [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Sub-dialog button](#)
 - [Dialog properties for a sub-dialog](#)
-

List Box

The List Box control allows you to display a list of items that support single or multiple selection and generate R script or Python for Spark script specific to the selected items. The List Box control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title that appears above the control. For multi-line titles, use \n to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

List Items. Click the ellipsis (...) button to open the [List Item Properties](#) dialog box, which allows you to specify the list items of the control. You can also open the List Item Properties dialog by double-clicking the List Box control on the canvas.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

List Box Type. Specifies whether the list box supports single selection only or multiple selection. You can also specify that items are displayed as a list of check boxes.

Separator Type. Specifies the delimiter between the selected list items in the generated script. The allowed separators are a blank, a comma, and a plus sign (+). You can also enter an arbitrary single character to be used as the separator.

Minimum Selected. The minimum number of items that must be selected in the control, if any.

Maximum Selected. The maximum number of items that can be selected in the control, if any.

Script. Specifies the R script or Python for Spark script that is generated by this control at run time and can be inserted in the script template.

- The value `%%ThisValue%%` specifies the run time value of the control and is the default. If the list items are manually defined, the run time value is the value of the Script property for the selected list item. If the list items are based on a target list control, the run time value is the value of the selected list item. For multiple selection list box controls, the run time value is a list of the selected items, delimited by

the specified Separator Type (default is blank separated). See the topic [Specifying list items for combo boxes and list boxes](#) for more information.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.

Quote Handling. Specifies handling of quotation marks in the run time value of `%%ThisValue%%` when the Script property contains `%%ThisValue%%` as part of a quoted string. In this context, a quoted string is a string that is enclosed in single quotation marks or double quotation marks. Quote handling applies only to quotation marks that are the same type as the quotation marks that enclose `%%ThisValue%%`. The following types of quote handling are available.

Python

Quotation marks in the run time value of `%%ThisValue%%` that match the enclosing quotation marks are escaped with the backslash character (\). For example, if the Script property is '`%%ThisValue%%`' and the selected list item is `List item's value`, then the generated script is '`List item\'s value`'. Note that quote handling is not done when `%%ThisValue%%` is enclosed in triple quotation marks.

R

Quotation marks in the run time value of `%%ThisValue%%` that match the enclosing quotation marks are escaped with the backslash character (\). For example, if the Script property is '`%%ThisValue%%`' and the selected list item is `List item's value`, then the generated script is '`List item\'s value`'.

None

Quotation marks in the run time value of `%%ThisValue%%` that match the enclosing quotation marks are retained with no modification.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Related information

- [Control types](#)
 - [Field Chooser](#)
 - [Filtering Field Lists](#)
 - [Check Box](#)
 - [Specifying list items for combo boxes and list boxes](#)
 - [Text control](#)
 - [Number Control](#)
 - [Static Text Control](#)
 - [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Sub-dialog button](#)
 - [Dialog properties for a sub-dialog](#)
-

Text control

The Text control is a simple text box that can accept arbitrary input, and has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title that appears above the control. For multi-line titles, use `\n` to specify line breaks.

Title Position. Specifies the position of the title relative to the control. Values are Top and Left where Top is the default.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

Text Content. Specifies whether the contents are arbitrary or whether the text box must contain a string that adheres to rules for IBM® SPSS® Modeler field names.

Default Value. The default contents of the text box.

Width. Specifies the width of the text area of the control in characters. The allowed values are positive integers. An empty value means that the width is automatically determined.

Required for execution. Specifies whether a value is required in this control in order for execution to proceed. If True is specified, the user of the node dialog must specify a value for the control otherwise clicking the OK button will generate an error. If False is specified, the absence of a value in this control has no effect on the state of the OK button. The default is False.

Script. Specifies the R script or Python for Spark script that is generated and run by this control at run time and can be inserted in the script template.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.
- The value `%%ThisValue%%` specifies the run time value of the control, which is the content of the text box. This is the default.
- If the Script property includes `%%ThisValue%%` and the run time value of the text box is empty, then the text box control does not generate any script.

Quote Handling. Specifies handling of quotation marks in the run time value of `%%ThisValue%%` when the Script property contains `%%ThisValue%%` as part of a quoted string. In this context, a quoted string is a string that is enclosed in single quotation marks or double quotation marks. Quote handling applies only to quotation marks that are the same type as the quotation marks that enclose `%%ThisValue%%`. The following types of quote handling are available.

Python

Quotation marks in the run time value of `%%ThisValue%%` that match the enclosing quotation marks are escaped with the backslash character (\). For example, if the Script property is '`%%ThisValue%%`' and the run time value of the text control is `Text box's value`, then the generated script is '`Text box\'s value`'. Quote handling is not done when `%%ThisValue%%` is enclosed in triple quotation marks.

R

Quotation marks in the run time value of `%%ThisValue%%` that match the enclosing quotation marks are escaped with the backslash character (\). For example, if the Script property is '`%%ThisValue%%`' and the run time value of the text control is `Text box's value`, then the generated script is '`Text box\'s value`'.

None

Quotation marks in the run time value of `%%ThisValue%%` that match the enclosing quotation marks are retained with no modification.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Related information

- [Control types](#)
 - [Field Chooser](#)
 - [Filtering Field Lists](#)
 - [Check Box](#)
 - [Combo Box](#)
 - [Specifying list items for combo boxes and list boxes](#)
 - [Number Control](#)
 - [Static Text Control](#)
 - [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Sub-dialog button](#)
 - [Dialog properties for a sub-dialog](#)
-

Number Control

The Number control is a text box for entering a numeric value, and has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title that appears above the control. For multi-line titles, use \n to specify line breaks.

Title Position. Specifies the position of the title relative to the control. Values are Top and Left where Top is the default.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

Numeric Type. Specifies any limitations on what can be entered. A value of Real specifies that there are no restrictions on the entered value, other than it be numeric. A value of Integer specifies that the value must be an integer.

Spin Input. Specifies whether the control is displayed as a spinner. The default is False.

Increment. The increment when the control is displayed as a spinner.

Default Value. The default value, if any.

Minimum Value. The minimum allowed value, if any.

Maximum Value. The maximum allowed value, if any.

Width. Specifies the width of the text area of the control in characters. The allowed values are positive integers. An empty value means that the width is automatically determined.

Required for execution. Specifies whether a value is required in this control in order for execution to proceed. If True is specified, the user of the node dialog must specify a value for the control otherwise clicking the OK button will generate an error. If False is specified, the absence of a value in this control has no effect on the state of the OK button. The default is False.

Script. Specifies the R script or Python for Spark script that is generated and run by this control at run time and can be inserted in the script template.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.
- The value `%%ThisValue%%` specifies the run time value of the control, which is the numeric value. This is the default.
- If the Script property includes `%%ThisValue%%` and the run time value of the number control is empty, then the number control does not generate any script.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Related information

- [Control types](#)
 - [Field Chooser](#)
 - [Filtering Field Lists](#)
 - [Check Box](#)
 - [Combo Box](#)
 - [Specifying list items for combo boxes and list boxes](#)
 - [Text control](#)
 - [Static Text Control](#)
 - [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Sub-dialog button](#)
 - [Dialog properties for a sub-dialog](#)
-

Date control

The Date control is a spinner control for specifying date/time values, which include dates, times, and datetimes. The Date control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title that appears above the control. For multi-line titles, use \n to specify line breaks.

Title Position. Specifies the position of the title relative to the control. Values are Top and Left where Top is the default.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

Type. Specifies whether the control is for dates, times, or datetime values.

Date

The control specifies a calendar date of the form yyyy-mm-dd. The default run time value is specified by the Default Value property.

Time

The control specifies the time of day in the form hh:mm:ss. The default run time value is the current time of day.

Datetime

The control specifies a date and time of the form yyyy-mm-dd hh:mm:ss. The default run time value is the current date and time of day.

Default Value. The default run time value of the control when the type is Date. You can specify to display the current date or a particular date.

Script. Specifies the R script or Python for Spark script that is generated and run by this control at run time and can be inserted in the script template.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.
- The value `%%ThisValue%%` specifies the run time value of the control. This is the default.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Note: The Date control is not supported in releases of IBM® SPSS® Modeler before release 18.

Related information

- [Control types](#)
 - [Field Chooser](#)
 - [Filtering Field Lists](#)
 - [Check Box](#)
 - [Combo Box](#)
 - [Specifying list items for combo boxes and list boxes](#)
 - [Number Control](#)
 - [Static Text Control](#)
 - [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Sub-dialog button](#)
 - [Dialog properties for a sub-dialog](#)
-

Secured Text

The Secured Text control is a text box that masks user entry with asterisks.

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title that appears above the control. For multi-line titles, use \n to specify line breaks.

Title Position. Specifies the position of the title relative to the control. Values are Top and Left where Top is the default.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

Width. Specifies the width of the text area of the control in characters. The allowed values are positive integers. An empty value means that the width is automatically determined.

Required for execution. Specifies whether a value is required in this control in order for execution to proceed. If True is specified, the user of the node dialog must specify a value for the control otherwise clicking the OK button will generate an error. If False is specified, the absence of a value in this control has no effect on the state of the OK button. The default is False.

Script. Specifies the R script or Python for Spark script that is generated and run by this control at run time and can be inserted in the script template.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.
- The value `%%ThisValue%%` specifies the run time value of the control, which is the content of the text box. This is the default.
- If the Script property includes `%%ThisValue%%` and the run time value of the secured text control is empty, then the secured text control does not generate any R script or Python for Spark script.

Quote Handling. Specifies handling of quotation marks in the run time value of `%%ThisValue%%` when the Script property contains `%%ThisValue%%` as part of a quoted string. In this context, a quoted string is a string that is enclosed in single quotation marks or double quotation marks. Quote handling applies only to quotation marks that are the same type as the quotation marks that enclose `%%ThisValue%%` and only when `Encrypt passed value=False`. The following types of quote handling are available.

Python

Quotation marks in the run time value of `%%ThisValue%%` that match the enclosing quotation marks are escaped with the backslash character (\). For example, if the Script property is '`%%ThisValue%%`' and the run time value of the control is `Secured Text's value`, then the generated script is '`Secured Text\'s value`'. Quote handling is not done when `%%ThisValue%%` is enclosed in triple quotation marks.

R

Quotation marks in the run time value of `%%ThisValue%%` that match the enclosing quotation marks are escaped with the backslash character (\). For example, if the Script property is '`%%ThisValue%%`' and the run time value of the control is `Secured Text's value`, then the generated script is '`Secured Text\'s value`'.

None

Quotation marks in the run time value of `%%ThisValue%%` that match the enclosing quotation marks are retained with no modification.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Note: The Secured Text control is not supported in releases of IBM® SPSS® Modeler before release 18.

Related information

- [Control types](#)
- [Field Chooser](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Combo Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Number Control](#)
- [Static Text Control](#)
- [Item Group](#)
- [Radio Group](#)
- [Defining radio buttons](#)
- [Check Box Group](#)
- [File Browser](#)
- [File type filter](#)
- [Tab](#)
- [Sub-dialog button](#)
- [Dialog properties for a sub-dialog](#)

Static Text Control

The Static Text control allows you to add a block of text to your node dialog, and has the following properties:

Identifier. The unique identifier for the control.

Title. The content of the text block. For multi-line content, use \n to specify line breaks.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Related information

- [Control types](#)
 - [Field Chooser](#)
 - [Filtering Field Lists](#)
 - [Check Box](#)
 - [Combo Box](#)
 - [Specifying list items for combo boxes and list boxes](#)
 - [Text control](#)
 - [Number Control](#)
 - [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Sub-dialog button](#)
 - [Dialog properties for a sub-dialog](#)
-

Color Picker

The Color Picker control is a user interface for specifying a color and generating the associated RGB value. The Color Picker control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title that appears above the control. For multi-line titles, use \n to specify line breaks.

Title Position. Specifies the position of the title relative to the control. Values are Top and Left where Top is the default.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

Script. Specifies the R script or Python for Spark script that is generated and run by this control at run time and can be inserted in the script template.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.
- The value %%**ThisValue**%% specifies the run time value of the control, which is the RGB value of the selected color. The RGB value is represented as a blank separated list of integers in the following order: R value, G value, B value.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Note: The Color Picker control is not supported in releases of IBM® SPSS® Modeler before release 18.

Related information

- [Control types](#)
- [Field Chooser](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Combo Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Number Control](#)
- [Static Text Control](#)

- [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Sub-dialog button](#)
 - [Dialog properties for a sub-dialog](#)
-

Table Control

The Table control creates a table with a fixed number of columns and a variable number of rows that are added at run time. The Table control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title that appears above the control. For multi-line titles, use \n to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

Reorder Buttons. Specifies whether move up and move down buttons are added to the table. These buttons are used at run time to reorder the rows of the table.

Table Columns. Click the ellipsis (...) button to open the [Table Columns](#) dialog box, where you specify the columns of the table.

Minimum Rows. The minimum number of rows that must be in the table.

Maximum Rows. The maximum number of rows that can be in the table.

Required for Execution. Specifies whether a value is required in this control in order for execution to proceed. If True is specified, the user of the node dialog must specify a value for the control otherwise clicking the OK button will generate an error. If False is specified, the absence of a value in this control has no effect on the state of the OK button.

Script. Specifies the R script or Python for Spark script that is generated by this control at run time and can be inserted in the script template.

- The value %%**ThisValue**%% specifies the run time value of the control and is the default. The run time value is a blank-separated list of the script that is generated by each column in the table, starting with the leftmost column. If the Script property includes %%**ThisValue**%% and none of the columns generate script, then the table as a whole does not generate any script.
- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Note: The Table control is not supported in releases of IBM® SPSS® Modeler before release 18.

- [Specifying columns for table controls](#)

Related information

- [Control types](#)
- [Field Chooser](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Text control](#)
- [Number Control](#)
- [Static Text Control](#)
- [Item Group](#)
- [Radio Group](#)
- [Defining radio buttons](#)
- [Check Box Group](#)
- [File Browser](#)

- [File type filter](#)
- [Tab](#)
- [Sub-dialog button](#)
- [Dialog properties for a sub-dialog](#)

Specifying columns for table controls

The Table Columns dialog box specifies the properties of the columns of the Table control.

Identifier. A unique identifier for the column.

Column Name. The name of the column as it appears in the table.

Contents. Specifies the type of data for the column. The value Real specifies that there are no restrictions on the entered value, other than it is numeric. The value Integer specifies that the value must be an integer. The value Any specifies that there are no restrictions on the entered value. The value Variable Name specifies that the value must meet the requirements for a valid variable name in IBM® SPSS® Statistics.

Default Value. The default value for this column, if any, when new rows are added to the table at run time.

Separator Type. Specifies the delimiter between the values of the column, in the generated script. The allowed separators are a blank, a comma, and a plus sign (+). You can also enter an arbitrary single character to be used as the separator.

Quoted. Specifies whether each value in the column is enclosed in double quotation marks in the generated script.

Quote Handling. Specifies handling of quotation marks in cell entries for the column when the Quoted property is true. Quote handling applies only to double quotation marks in cell values. The following types of quote handling are available.

Python

Double quotation marks in cell values are escaped with the backslash character (\). For example, if the cell value is `This "quoted" value` then the generated script is `"This \\\"quoted\\\" value"`.

R

Double quotation marks in cell values are escaped with the backslash character (\). For example, if the cell value is `This "quoted" value` then the generated script is `"This \\\"quoted\\\" value"`.

None

Double quotation marks in cell values are retained with no modification.

Width(chars). Specifies the width of the column in characters. The allowed values are positive integers.

Script. Specifies the R script or Python for Spark script that is generated by this column at run time. The generated script for the table as a whole is a blank-separated list of the script that is generated by each column in the table, starting with the leftmost column.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.
- The value `%%ThisValue%%` specifies the run time value of the column, which is a list of the values in the column, delimited by the specified separator.
- If the Script property for the column includes `%%ThisValue%%` and the run time value of the column is empty, then the column does not generate any script.

Note: You can add a row for a new Table column in the blank line at the bottom of the existing list in the Table Columns dialog. Entering any of the properties other than the identifier generates a unique identifier, which you can keep or modify. You can delete a Table column by clicking the identifier cell for the Table column and pressing delete.

Link to Control

You can link a Table control to a Field Chooser control. When a Table control is linked to a Field Chooser, there is a row in the table for each field in the Field Chooser. Rows are added to the table by adding fields to the Field Chooser. Rows are deleted from the table by removing fields from the Field Chooser. A linked Table control can be used, for example, to specify properties of fields that are selected in a Field Chooser.

To enable linking, the table must have a column with Variable Name for the Contents property and there must be at least one Field Chooser control on the canvas.

To link a Table control to a Field Chooser, specify the Field Chooser from the list of Available controls in the Link to Control group on the Table Columns dialog box. Then, select the table column, referred to as the Linked column, that defines the link. When the table is rendered, the linked column displays the current fields in the Field Chooser. You can link only to multi-field Field Choosers.

Related information

- [Control types](#)
 - [Field Chooser](#)
 - [Filtering Field Lists](#)
 - [Check Box](#)
 - [Combo Box](#)
 - [Text control](#)
 - [Number Control](#)
 - [Static Text Control](#)
 - [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Sub-dialog button](#)
 - [Dialog properties for a sub-dialog](#)
-

Item Group

The Item Group control is a container for other controls, allowing you to group and control the script generated from multiple controls. For example, you have a set of check boxes that specify optional settings for a subcommand, but only want to generate the script for the subcommand if at least one box is checked. This is accomplished by using an Item Group control as a container for the check box controls. The following types of controls can be contained in an Item Group: field chooser, check box, combo box, list box, text control, number control, static text, radio group, and file browser. The Item Group control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title for the group. For multi-line titles, use \n to specify line breaks.

Script. Specifies the R script or Python for Spark script that is generated and run by this control at run time and can be inserted in the script template.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.
- You can include identifiers for any controls contained in the item group. At run time the identifiers are replaced with the R script or Python script generated by the controls.
- The value %%**ThisValue**%% generates a blank-separated list of the R script or Python script generated by each control in the item group, in the order in which they appear in the group (top to bottom). This is the default. If the Script property includes %%**ThisValue**%% and no script is generated by any of the controls in the item group, then the item group as a whole does not generate any script.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Related information

- [Control types](#)
- [Field Chooser](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Combo Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Text control](#)
- [Number Control](#)
- [Static Text Control](#)
- [Radio Group](#)
- [Defining radio buttons](#)
- [Check Box Group](#)
- [File Browser](#)
- [File type filter](#)
- [Tab](#)
- [Sub-dialog button](#)
- [Dialog properties for a sub-dialog](#)

Radio Group

The Radio Group control is a container for a set of radio buttons, each of which can contain a set of nested controls. The Radio Group control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title for the group. For multi-line titles, use `\n` to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Radio Buttons. Click the ellipsis (...) button to open the [Radio Group Properties](#) dialog box, which allows you to specify the properties of the radio buttons as well as to add or remove buttons from the group. The ability to nest controls under a given radio button is a property of the radio button and is set in the Radio Group Properties dialog box. Note that you can also open the Radio Group Properties dialog by double-clicking the Radio Group control on the canvas.

Script. Specifies the R script or Python for Spark script that is generated by this control at run time and can be inserted in the script template.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.
- The value `%%ThisValue%%` specifies the run time value of the radio button group, which is the value of the Script property for the selected radio button. This is the default. If the Script property includes `%%ThisValue%%` and no script is generated by the selected radio button, then the radio button group does not generate any script.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

- [Defining radio buttons](#)

Related information

- [Control types](#)
- [Field Chooser](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Combo Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Text control](#)
- [Number Control](#)
- [Static Text Control](#)
- [Item Group](#)
- [Defining radio buttons](#)
- [Check Box Group](#)
- [File Browser](#)
- [File type filter](#)
- [Tab](#)
- [Sub-dialog button](#)
- [Dialog properties for a sub-dialog](#)

Defining radio buttons

The Radio Button Group Properties dialog box allows you to specify a group of radio buttons.

Identifier. A unique identifier for the radio button.

Column Name. The name that appears next to the radio button. The name is a required field.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the name to use as a mnemonic. The specified character must exist in the name.

Nested Group. Specifies whether other controls can be nested under this radio button. The default is false. When the nested group property is set to true, a rectangular drop zone is displayed, nested and indented, under the associated radio button. The following controls can be nested under a radio button: field chooser, check box, text control, static text, number control, combo box, list box, and file browser.

Default. Specifies whether the radio button is the default selection.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Script. Specifies the R script or Python for Spark script that is generated when the radio button is selected.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.
- For radio buttons containing nested controls, the value `%%ThisValue%%` generates a blank-separated list of the R script or Python for Spark script generated by each nested control, in the order in which they appear under the radio button (top to bottom).

You can add a new radio button in the blank line at the bottom of the existing list. Entering any of the properties other than the identifier will generate a unique identifier, which you can keep or modify. You can delete a radio button by clicking on the *Identifier* cell for the button and pressing delete.

Related information

- [Control types](#)
- [Field Chooser](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Combo Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Text control](#)
- [Number Control](#)
- [Static Text Control](#)
- [Item Group](#)
- [Radio Group](#)
- [Check Box Group](#)
- [File Browser](#)
- [File type filter](#)
- [Tab](#)
- [Sub-dialog button](#)
- [Dialog properties for a sub-dialog](#)

Check Box Group

The Check Box Group control is a container for a set of controls that are enabled or disabled as a group, by a single check box. The following types of controls can be contained in a Check Box Group: field chooser, check box, combo box, list box, text control, number control, static text, radio group, and file browser. The Check Box Group control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title for the group. For multi-line titles, use `\n` to specify line breaks.

Checkbox Title. An optional label that is displayed with the controlling check box. Supports `\n` to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

Default Value. The default state of the controlling check box--checked or unchecked.

Checked/Unchecked R Script. Specifies the R script that is generated when the control is checked and when it is unchecked. To include the R script in the script template, use the value of the Identifier property. The generated R script, whether from the Checked R Script or Unchecked R Script property, will be inserted at the specified positions of the identifier. For example, if the identifier is `checkboxgroup1`, then at run time, instances of `%%checkboxgroup1%%` in the script template will be replaced by the value of the Checked R Script property when the box is checked and the Unchecked R Script property when the box is unchecked.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.
- You can include identifiers for any controls contained in the check box group. At run time the identifiers are replaced with the R script generated by the controls.

- The value `%%ThisValue%%` can be used in either the Checked R Script or Unchecked R Script property. It generates a blank-separated list of the R script generated by each control in the check box group, in the order in which they appear in the group (top to bottom).
- By default, the Checked R Script property has a value of `%%ThisValue%%` and the Unchecked R Script property is blank.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Related information

- [Control types](#)
- [Field Chooser](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Combo Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Text control](#)
- [Number Control](#)
- [Static Text Control](#)
- [Item Group](#)
- [Radio Group](#)
- [Defining radio buttons](#)
- [File Browser](#)
- [File type filter](#)
- [Tab](#)
- [Sub-dialog button](#)
- [Dialog properties for a sub-dialog](#)

File Browser

The File Browser control consists of a text box for a file path and a browse button that opens a standard IBM® SPSS® Modeler dialog to open or save a file. The File Browser control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the script template.

Title. An optional title that appears above the control. For multi-line titles, use `\n` to specify line breaks.

Title Position. Specifies the position of the title relative to the control. Values are Top and Left where Top is the default.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

File System Operation. Specifies whether the dialog launched by the browse button is appropriate for opening files or for saving files. A value of Open indicates that the browse dialog validates the existence of the specified file. A value of Save indicates that the browse dialog does not validate the existence of the specified file.

Browser Type. Specifies whether the browse dialog is used to select a file (Locate File) or to select a folder (Locate Folder).

File Filter. Click the ellipsis (...) button to open the [File Filter](#) dialog box, which allows you to specify the available file types for the open or save dialog. By default, all file types are allowed. Note that you can also open the File Filter dialog by double-clicking the File Browser control on the canvas.

File System Type. In distributed analysis mode, this specifies whether the open or save dialog browses the file system on which IBM SPSS Modeler Server is running or the file system of your local computer. Select Server to browse the file system of the server or Client to browse the file system of your local computer. The property has no effect in local analysis mode.

Required for execution. Specifies whether a value is required in this control in order for execution to proceed. If True is specified, the user of the node dialog must specify a value for the control otherwise clicking the OK button will generate an error. If False is specified, the absence of a value in this control has no effect on the state of the OK button. The default is False.

Default. The default value of the control.

Script. Specifies the R script or Python for Spark script that is generated by this control at run time and can be inserted in the script template.

- You can specify any valid R script or Python for Spark script. For multi-line scripts or long scripts, click the ellipsis (...) button and enter your script in the Script Property dialog.

- The value `%%ThisValue%%` specifies the run time value of the text box, which is the file path enclosed by double quotation marks, specified manually or populated by the browse dialog. This is the default.
- If the Script property includes `%%ThisValue%%` and the run time value of the text box is empty, then the file browser control does not generate any script.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

- [File type filter](#)

Related information

- [Control types](#)
- [Field Chooser](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Combo Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Text control](#)
- [Number Control](#)
- [Static Text Control](#)
- [Item Group](#)
- [Radio Group](#)
- [Defining radio buttons](#)
- [Check Box Group](#)
- [File type filter](#)
- [Tab](#)
- [Sub-dialog button](#)
- [Dialog properties for a sub-dialog](#)

File type filter

The File Filter dialog box allows you to specify the file types displayed in the Files of type and Save as type drop-down lists for open and save dialogs accessed from a File System Browser control. By default, all file types are allowed.

To specify file types not explicitly listed in the dialog box:

1. Select Other.
2. Enter a name for the file type.
3. Enter a file type using the form `* .suffix`--for example, `* .xls`. You can specify multiple file types, each separated by a semicolon.

Related information

- [Control types](#)
- [Field Chooser](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Combo Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Text control](#)
- [Number Control](#)
- [Static Text Control](#)
- [Item Group](#)
- [Radio Group](#)
- [Defining radio buttons](#)
- [Check Box Group](#)
- [File Browser](#)
- [Tab](#)
- [Sub-dialog button](#)
- [Dialog properties for a sub-dialog](#)

Tab

The Tab control adds a tab to the node dialog. Any of the other controls can be added to the new tab. The Tab control has the following properties:

Identifier. The unique identifier for the control.

Title. The title of the tab.

Position. Specifies the position of the tab on the node dialog, relative to the other tabs on the node dialog.

Script. Specifies the R script or Python for Spark script that is generated and run by this control at run time and can be inserted in the script template.

- You can specify any valid R script or Python for Spark script and you can use `\n` for line breaks.
- The value `%%ThisValue%%` generates a blank-separated list of the R script or Python for Spark script generated by each control in the tab, in the order in which they appear on the tab (top to bottom and left to right). This is the default.
- If the Script property includes `%%ThisValue%%` and no R script or Python for Spark script is generated by any of the controls in the tab, then the tab as a whole does not generate any script.

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Related information

- [Control types](#)
- [Field Chooser](#)
- [Filtering Field Lists](#)
- [Check Box](#)
- [Combo Box](#)
- [Specifying list items for combo boxes and list boxes](#)
- [Text control](#)
- [Number Control](#)
- [Static Text Control](#)
- [Item Group](#)
- [Radio Group](#)
- [Defining radio buttons](#)
- [Check Box Group](#)
- [File Browser](#)
- [File type filter](#)
- [Sub-dialog button](#)
- [Dialog properties for a sub-dialog](#)

Sub-dialog button

The Sub-dialog Button control specifies a button for launching a sub-dialog and provides access to the Dialog Builder for the sub-dialog. The Sub-dialog Button has the following properties:

Identifier. The unique identifier for the control.

Title. The text that is displayed in the button.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Sub-dialog. Click the ellipsis (...) button to open the Custom Dialog Builder for the sub-dialog. You can also open the builder by double-clicking on the Sub-dialog button.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key].

Enabling Rule. Specifies a rule that determines when the current control is enabled. Click the ellipsis (...) button to open the [Enabling Rule](#) dialog box and specify the rule. The Enabling Rule property is visible only when other controls that can be used to specify an enabling rule exist on the canvas.

Note: The Sub-dialog Button control cannot be added to a sub-dialog.

- [Dialog properties for a sub-dialog](#)

Related information

- [Control types](#)
 - [Field Chooser](#)
 - [Filtering Field Lists](#)
 - [Check Box](#)
 - [Combo Box](#)
 - [Specifying list items for combo boxes and list boxes](#)
 - [Text control](#)
 - [Number Control](#)
 - [Static Text Control](#)
 - [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Dialog properties for a sub-dialog](#)
-

Dialog properties for a sub-dialog

To view and set properties for a sub-dialog:

1. Open the sub-dialog by double-clicking on the button for the sub-dialog in the main dialog, or single-click the sub-dialog button and click the ellipsis (...) button for the Sub-dialog property.
2. In the sub-dialog, click on the canvas in an area outside of any controls. With no controls on the canvas, the properties for a sub-dialog are always visible.

Sub-dialog Name. The unique identifier for the sub-dialog. The Sub-dialog Name property is required.

Note: If you specify the Sub-dialog Name as an identifier in the Script Template--as in `%%My Sub-dialog`

`Name%%`--it will be replaced at run-time with a blank-separated list of the script generated by each control in the sub-dialog, in the order in which they appear (top to bottom and left to right).

Title. Specifies the text to be displayed in the title bar of the sub-dialog box. The Title property is optional but recommended.

Help File. Specifies the path to an optional help file for the sub-dialog. This is the file that will be launched when the user clicks the Help button on the sub-dialog, and may be the same help file specified for the main dialog. Help files must be in HTML format. See the description of the Help File property for [Dialog Properties](#) for more information.

Related information

- [Control types](#)
 - [Field Chooser](#)
 - [Filtering Field Lists](#)
 - [Check Box](#)
 - [Combo Box](#)
 - [Specifying list items for combo boxes and list boxes](#)
 - [Text control](#)
 - [Number Control](#)
 - [Static Text Control](#)
 - [Item Group](#)
 - [Radio Group](#)
 - [Defining radio buttons](#)
 - [Check Box Group](#)
 - [File Browser](#)
 - [File type filter](#)
 - [Tab](#)
 - [Sub-dialog button](#)
-

Specifying an Enabling Rule for a Control

You can specify a rule that determines when a control is enabled. For example, you can specify that a radio group is enabled when a field chooser is populated. The available options for specifying the enabling rule depend on the type of control that defines the rule.

You can specify that the current control is enabled when a Field Chooser is populated with at least one field (Non-empty). You can alternately specify that the current control is enabled when a Field Chooser is not populated (Empty).

Check Box or Check Box Group

You can specify that the current control is enabled when a Check Box or Check Box Group is checked. You can alternately specify that the current control is enabled when a Check Box or Check Box Group is not checked.

Combo Box or Single-Select List Box

You can specify that the current control is enabled when a particular value is selected in a Combo Box or Single-Select List Box. You can alternately specify that the current control is enabled when a particular value is not selected in a Combo Box or Single-Select List Box.

Multi-Select List Box

You can specify that the current control is enabled when a particular value is among the selected values in a Multi-Select List Box. You can alternately specify that the current control is enabled when a particular value is not among the selected values in a Multi-Select List Box.

Radio Group

You can specify that the current control is enabled when a particular radio button is selected. You can alternately specify that the current control is enabled when a particular radio button is not selected.

Controls for which an enabling rule can be specified have an associated Enabling Rule property.

Note:

- Enabling rules apply whether the control that defines the rule is enabled or not. For example, consider a rule that specifies that a radio group is enabled when a field chooser is populated. The radio group is then enabled whenever the field chooser is populated, regardless of whether the field chooser is enabled.
- When a Tab control is disabled, all controls on the tab are disabled regardless of whether any of those controls have an enabling rule that is satisfied.
- When a Check Box Group is disabled, all controls in the group are disabled regardless of whether the controlling check box is checked.

Extension Properties

The Extension Properties dialog specifies information about the current extension within the Custom Dialog Builder for Extensions, such as the name of the extension and the files in the extension.

- All custom node dialogs that are created in the Custom Dialog Builder for Extensions are part of an extension.
- Fields on the Required tab of the Extension Properties dialog must be specified before you can install an extension and the custom node dialogs that are contained in it.

To specify the properties for an extension, from the menus in the Custom Dialog Builder for Extensions choose:

Extension > Properties

- [Required properties of extensions](#)
- [Optional properties of extensions](#)

Required properties of extensions

Name

A unique name to associate with the extension. It can consist of up to three words and is not case sensitive. Characters are restricted to seven-bit ASCII. To minimize the possibility of name conflicts, you may want to use a multi-word name, where the first word is an identifier for your organization, such as a URL. Note that name will also be used for the extension bundle (.mpe) file name by default when you save the extension. We recommend using the default name when you save. If you save with a different name, you will not be able to uninstall the extension in the future.

Summary

A short description of the extension that is intended to be displayed as a single line.

Version

A version identifier of the form x.x.x, where each component of the identifier must be an integer, as in 1.0.0. Zeros are implied if not provided. For example, a version identifier of 3.1 implies 3.1.0. The version identifier is independent of the IBM® SPSS® Modeler version.

Minimum SPSS Modeler Version

The minimum version of SPSS Modeler required to run the extension.

Files

The Files list displays the files that are currently included in the extension. Click Add to add files to the extension. You can also remove files from the extension and you can extract files to a specified folder.

- Custom node dialogs have a filetype of .cfe.
- Translation files for components of the extension are added from the Localization settings on the Optional tab.
- You can add a readme file to the extension. Specify the filename as ReadMe.txt. Users can access the readme file from the dialog that displays the details for the extension. You can include localized versions of the readme file, specified as ReadMe_<language>

Optional properties of extensions

General properties

Description

A more detailed description of the extension than that provided for the Summary field. For example, you might list the major features available with the extension.

Date

An optional date for the current version of the extension. No formatting is provided.

Author

The author of the extension. You might want to include an email address.

Links

A set of URLs to associate with the extension; for example, the author's home page. The format of this field is arbitrary so be sure to delimit multiple URLs with spaces, commas, or some other reasonable delimiter.

Keywords

A set of keywords with which to associate the extension.

Platform

Information about any restrictions that apply to using the extension on particular operating system platforms.

Dependencies

Maximum SPSS Modeler Version

The maximum version of IBM® SPSS® Modeler on which the extension can be run.

Integration Plug-in for R required

Specifies whether the Integration Plug-in for R is required.

If the extension requires any R packages from the CRAN package repository, then enter the names of those packages in the Required R Packages control. Names are case sensitive. To add the first package, click anywhere in the Required R Packages control to highlight the entry field. Pressing Enter, with the cursor in a given row, will create a new row. You delete a row by selecting it and pressing Delete.

Localization

Custom Nodes

You can add translated versions of the properties file (specifies all strings that appear in the node dialog) for a custom node dialog within the extension. To add translations for a particular node dialog, select the dialog, click Add Translations, and select the folder that contains the translated versions. All translated files for a particular node dialog must be in the same folder. For instructions on creating the translation files, see the topic [Creating Localized Versions of Custom Node Dialogs](#).

Translation Catalogues Folder

You can provide localized versions of the Summary and Description fields for the extension that are displayed when end users view the extension details from the Extension Hub. The set of all localized files for an extension must be in a folder named lang. Browse to the lang folder that contains the localized files and select that folder.

To provide localized versions of the Summary and Description fields, create a file named <extension name>_<language-identifier>.properties for each language for which a translation is being provided. At run time, if the .properties file for the current user interface language cannot be found, the values of the Summary and Description fields that are specified on the Required and Optional tabs are used.

- <extension name> is the value of the Name field for the extension, with any spaces replaced by underscore characters.
- <language-identifier> is the identifier for a particular language. Identifiers for the languages that are supported by IBM SPSS Modeler are shown in what follows.

For example, the French translations for an extension named MYORG MYSTAT are stored in the file MYORG_MYSTAT_fr.properties.

The .properties file must contain the following two lines, which specify the localized text for the two fields:

```
Summary=<localized text for Summary field>
Description=<localized text for Description field>
```

- The keywords Summary and Description must be in English and the localized text must be on the same line as the keyword, and with no line breaks.
- The file must be in ISO 8859-1 encoding. Characters that cannot be directly represented in this encoding must be written with Unicode escapes ("\\u").

The lang folder that contains the localized files must have a subfolder named <language-identifier> that contains the localized .properties file for a particular language. For example, the French .properties file must be in the lang/fr folder.

Language Identifiers

de. German

en. English

es. Spanish

fr. French

it. Italian

ja. Japanese

ko. Korean

pl. Polish

pt_BR. Brazilian Portuguese

ru. Russian

zh_CN. Simplified Chinese

zh_TW. Traditional Chinese

Managing custom node dialogs

The Custom Dialog Builder for Extensions allows you to manage custom node dialogs, within extensions that are created by you or by other users. Custom node dialogs must be installed to all instances of SPSS® Modeler Client or SPSS Modeler Batch where they are needed before they can be used. Note that there is nothing that needs to be installed to SPSS Modeler Server to use a custom dialog node in server mode.

Note: You can only modify custom node dialogs that are created in IBM® SPSS Modeler.

Opening an extension that contains custom node dialogs

You can open an extension bundle file (.mpe) that contains the specifications for one or more custom node dialogs or you can open an installed extension. You can modify any of the node dialogs in the extension and save or install the extension. Installing the extension installs the node dialogs that are contained in the extension. Saving the extension saves changes that were made to any of the node dialogs in the extension.

To open an extension bundle file, from the menus in the Custom Dialog Builder for Extensions choose:

File > Open

To open an installed extension, from the menus in the Custom Dialog Builder for Extensions choose:

File > Open Installed

Note: If you are opening an installed extension to modify it, choosing File > Install reinstalls it, replacing the existing version. Using Edit on the context menu of a node created using the Custom Dialog Builder will not open the node dialog in the Custom Dialog Builder.

Saving to an extension bundle file

Saving the extension that is open in the Custom Dialog Builder for Extensions also saves the custom node dialogs that are contained in the extension. Extensions are saved to an extension bundle file (.mpe). We recommend keeping the default file name, which matches the name you specified in Name field of the Extension Properties dialog.

From the menus in the Custom Dialog Builder for Extensions, choose:

File > Save

Installing an extension

Installing the extension that is open in the Custom Dialog Builder for Extensions also installs the custom node dialogs that are contained in the extension. Installing an existing extension replaces the existing version, which includes replacing all custom node dialogs in the extension that were already installed.

To install the currently open extension, from the menus in the Custom Dialog Builder for Extensions choose:

File > Install

By default, extensions are installed to a general user-writable location for your operating system. For more information, see the topic [Installation locations for extensions](#).

Note: In an open stream, the existing versions of the node dialogs that are contained in the extension will not be replaced. When you open a stream that contains a Custom Dialog Builder node that has been re-installed, you will receive a warning message.

Uninstalling an extension

From the menus in the Custom Dialog Builder for Extensions, choose:

File > Uninstall

Uninstalling an extension uninstalls all custom node dialogs that are contained in the extension. You can also uninstall extensions from the Extension Hub.

Note: To successfully uninstall an extension, the extension bundle .mpe file name must match the name specified in the Extension Properties dialog. This is the default file name. If you modified the file name, rename it to match the Name field and try uninstalling again.

Importing a custom dialog package file

You can import a custom dialog package (.cdf) file into the Custom Dialog Builder for Extensions. The .cdf file is converted into a .cfi file, which is added to a new extension.

From the menus in the Custom Dialog Builder for Extensions, choose:

File > Import

You can also add .cfi files to an extension from the Extension Properties dialog, which is accessed from Extension > Properties within the Custom Dialog Builder for Extensions.

Adding a custom node dialog to an extension

You can add a new custom node dialog to an extension.

From the menus in the Custom Dialog Builder for Extensions, choose:

Extension > New Dialog

Switching between multiple custom node dialogs in an extension

If the current extension contains multiple custom node dialogs, you can switch between them.

From the menus in the Custom Dialog Builder for Extensions, choose:

Extension > Edit Dialog and select the custom node dialog that you want to work with.

Creating a new extension

When you create a new extension in the Custom Dialog Builder for Extensions, a new empty custom node dialog is added to the extension.

To create a new extension, from the menus in the Custom Dialog Builder for Extensions choose:

File > New

Extensions in SPSS Modeler Batch or in IBM SPSS Collaboration and Deployment Services

To use extensions in an SPSS Modeler Batch or IBM SPSS Collaboration and Deployment Services installation, ensure that the environment variable *IBM_SPSS_MODELER_EXTENSION_PATH* is defined on the target environment, and that it points to the location that contains the extensions. If the streams that contain a custom node were stored to the IBM SPSS Collaboration and Deployment Services Repository before the *IBM_SPSS_MODELER_EXTENSION_PATH* environment variable was defined, you must re-store the streams to the repository before they will run successfully.

Note: Ensure that the version of the SPSS Modeler Batch or IBM SPSS Collaboration and Deployment Services adapter for SPSS Modeler matches the version of SPSS Modeler Client where the extension was created.

Creating Localized Versions of Custom Node Dialogs

You can create localized versions of custom node dialogs for any of the languages supported by IBM® SPSS® Modeler. You can localize any string that appears in a custom node dialog and you can localize the optional help file.

To localize dialog strings

You must create a copy of the properties file associated with the custom node dialog for each language that you plan to deploy. The properties file contains all of the localizable strings associated with the node dialog.

Extract the custom node dialog file (.cfe) from the extension by selecting the file in the Extension Properties dialog (within the Custom Dialog Builder for Extensions) and clicking Extract. Then, extract the contents of the .cfe file. A .cfe file is simply a .zip file. The extracted contents of a .cfe file includes a properties file for each supported language, where the name of the file for a particular language is given by <Dialog Name>_<language identifier>.properties (see language identifiers in the table that follows).

1. Open each properties file, that you plan to translate, with a text editor that supports UTF-8, such as Notepad on Windows. Modify the values associated with any properties that need to be localized, but do not modify the names of the properties. Properties associated with a specific control are prefixed with the identifier for the control. For example, the ToolTip property for a control with the identifier *options_button* is *options_button_tooltip_LABEL*. Title properties are simply named <*identifier*>_LABEL, as in *options_button_LABEL*.
2. Add the localized versions of the properties files back to the custom node dialog file (.cfe) from the Localization settings on the Optional tab of the Extension Properties dialog. For more information, see the topic [Optional properties of extensions](#).

When the node dialog is launched, IBM SPSS Modeler searches for a properties file whose language identifier matches the current language, as specified by the Language drop-down on the General tab in the Options dialog box. If no such properties file is found, the default file <Dialog Name>.properties is used.

To localize the help file

1. Make a copy of the help file associated with the custom node dialog and localize the text for the language you want.
2. Rename the copy to <Help File>_<language identifier>.htm, using the language identifiers in the table below. For example, if the help file is myhelp.htm and you want to create a German version of the file, then the localized help file should be named myhelp_de.htm.

Store all localized versions of the help file in the same directory as the non-localized version. When you add the non-localized help file from the Help File property of Dialog Properties, the localized versions are automatically added to the node dialog.

If there are supplementary files such as image files that also need to be localized, then you must manually modify the appropriate paths in the main help file to point to the localized versions. Supplementary files, including localized versions, must be manually added to the custom node dialog (.cfe) file. See the previous section titled "To localize dialog strings" for information about accessing and manually modifying custom node dialog files.

When the node dialog is launched, IBM SPSS Modeler searches for a help file whose language identifier matches the current language, as specified by the Language drop-down on the General tab in the Options dialog box. If no such help file is found, the help file specified for the node dialog (the file specified in the Help File property of Dialog Properties) is used.

Language Identifiers

de. German

en. English

es. Spanish

fr. French

it. Italian

ja. Japanese

ko. Korean

pl. Polish

pt_BR. Brazilian Portuguese

ru. Russian

zh_CN. Simplified Chinese

zh_TW. Traditional Chinese

Note: Text in custom node dialogs and associated help files is not limited to the languages supported by IBM SPSS Modeler. You are free to write the node dialog and help text in any language without creating language-specific properties and help files. All users of your node dialog will then see the text in that language.

Related information

- [Creating and managing custom nodes](#)
 - [Custom Dialog Builder layout](#)
 - [Building a custom node dialog](#)
 - [Dialog Properties](#)
 - [Laying out controls on the dialog canvas](#)
 - [Building the script template](#)
 - [Previewing a custom node dialog](#)
 - [Managing custom node dialogs](#)
 - [Control types](#)
-

Importing and exporting data using Python for Spark

Using the Custom Dialog Builder for Extensions, you can create custom nodes and write Python for Spark scripts to read data from wherever your data source is, and write data out to any data format supported by Apache Spark.

For example, a user wants to write his data to a database. He uses the Custom Dialog Builder for Extensions and Python for Spark to create a custom export JDBC node and then runs the model to write data into a database. To read data from the database, he can also create a custom import JDBC node. He could also use this same method to read data into SPSS® Modeler from a JSON file, for example. Then, after reading his data into SPSS Modeler, he can use all available SPSS Modeler nodes to work on his business problem.

Note: If you want to use JDBC with Python for Spark import and export functionality, you must copy your JDBC driver file to the as/lib directory inside your IBM® SPSS Modeler installation directory.

To import/export data using Python for Spark

1. Go to Extensions...>Custom Node Dialog Builder.
 2. Under Dialog Properties, select Python for Spark for the Script Type and select Import or Export for the Node Type.
 3. Enter other properties as desired, such as a Dialog Name.
 4. In the Script section, type or paste your Python for Spark script for importing or exporting data.
 5. Click Install to install the Python for Spark script. New custom import nodes will be added to the Sources palette, and new custom export nodes will be added to the Export palette.
-

Importing and exporting data using R

Using the Custom Dialog Builder for Extensions, you can create custom nodes and write R scripts to read data from wherever your data source is, and write data out to any data format supported by R.

For example, a user wants to write her data to a database. She uses the Custom Dialog Builder for Extensions and R scripting to create a custom export JDBC node and then runs the model to write data into a database. To read data from the database, she can also create a custom import JDBC node. She could also use this same method to read data into SPSS® Modeler from a JSON file, for example. Then, after reading his data into SPSS Modeler, he can use all available SPSS Modeler nodes to work on his business problem.

To import/export data using R

1. Go to Extensions...>Custom Node Dialog Builder.
 2. Under Dialog Properties, select R for the Script Type and select Import or Export for the Node Type.
 3. Enter other properties as desired, such as a Dialog Name.
 4. In the Script section, type or paste your R script for importing or exporting data.
 5. Click Install to install the R script. New custom import nodes will be added to the Sources palette, and new custom export nodes will be added to the Export palette.
-

Introduction to CRISP-DM

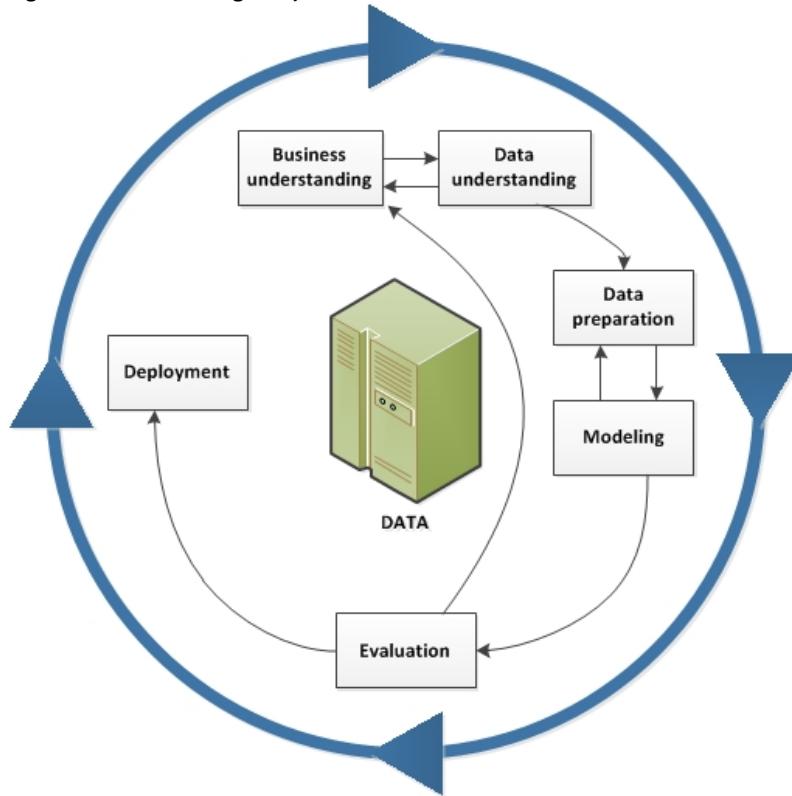
- [CRISP-DM Help Overview](#)

CRISP-DM Help Overview

CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining, is an industry-proven way to guide your data mining efforts.

- As a **methodology**, it includes descriptions of the typical phases of a project, the tasks involved with each phase, and an explanation of the relationships between these tasks.
- As a **process model**, CRISP-DM provides an overview of the data mining life cycle.

Figure 1. The data mining life cycle



The life cycle model consists of six phases with arrows indicating the most important and frequent dependencies between phases. The sequence of the phases is not strict. In fact, most projects move back and forth between phases as necessary.

The CRISP-DM model is flexible and can be customized easily. For example, if your organization aims to detect money laundering, it is likely that you will sift through large amounts of data without a specific modeling goal. Instead of modeling, your work will focus on data exploration and visualization to uncover suspicious patterns in financial data. CRISP-DM allows you to create a data mining model that fits your particular needs.

In such a situation, the modeling, evaluation, and deployment phases might be less relevant than the data understanding and preparation phases. However, it is still important to consider some of the questions raised during these later phases for long-term planning and future data mining goals.

- [CRISP-DM in IBM SPSS Modeler](#)
- [Additional resources](#)

CRISP-DM in IBM SPSS Modeler

IBM® SPSS® Modeler incorporates the CRISP-DM methodology in two ways to provide unique support for effective data mining.

- The CRISP-DM project tool helps you organize project streams, output, and annotations according to the phases of a typical data mining project. You can produce reports at any time during the project based on the notes for streams and CRISP-DM phases.
- Help for CRISP-DM guides you through the process of conducting a data mining project. The help system includes tasks lists for each step as well as examples of how CRISP-DM works in the real world. You can access CRISP-DM Help by choosing CRISP-DM Help from the main window Help menu.

- [CRISP-DM Project Tool](#)
- [Help for CRISP-DM](#)

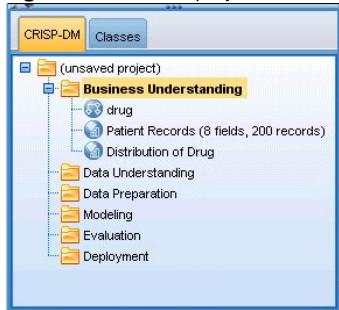
Related information

- [Additional resources](#)

CRISP-DM Project Tool

The CRISP-DM project tool provides a structured approach to data mining that can help ensure your project's success. It is essentially an extension of the standard IBM® SPSS® Modeler project tool. In fact, you can toggle between the CRISP-DM view and the standard Classes view to see your streams and output organized by type or by phases of CRISP-DM.

Figure 1. CRISP-DM project tool



Using the CRISP-DM view of the project tool, you can:

- Organize a project's streams and output according to data mining phases.
- Take notes on your organization's goals for each phase.
- Create custom tooltips for each phase.
- Take notes on the conclusions drawn from a particular graph or model.
- Generate an HTML report or update for distribution to the project team.

Related information

- [Additional resources](#)
- [Help for CRISP-DM](#)

Help for CRISP-DM

IBM® SPSS® Modeler offers an online guide for the non-proprietary CRISP-DM process model. The guide is organized by project phases and provides the following support:

- An overview and task list for each phase of CRISP-DM
- Help on producing reports for various milestones
- Real-world examples illustrating how a project team can use CRISP-DM to light the way for data mining
- Links to additional resources on CRISP-DM

You can access CRISP-DM Help by choosing CRISP-DM Help from the main window Help menu.

Related information

- [Additional resources](#)
- [CRISP-DM Help Overview](#)

Additional resources

In addition to IBM® SPSS® Modeler support for CRISP-DM, there are several ways to expand your understanding of data mining processes.

- Read the CRISP-DM manual, created by the CRISP-DM consortium and supplied with this release.
 - Read *Data Mining with Confidence*, copyright 2002 by SPSS Inc., ISBN 1-56827-287-1.
-

Business Understanding

- [Business Understanding Overview](#)
 - [Determining Business Objectives](#)
 - [Assessing the Situation](#)
 - [Determining Data Mining Goals](#)
 - [Producing a Project Plan](#)
 - [Ready for the next step?](#)
-

Business Understanding Overview

Even before working in IBM® SPSS® Modeler, you should take the time to explore what your organization expects to gain from data mining. Try to involve as many key people as possible in these discussions and document the results. The final step of this CRISP-DM phase discusses how to produce a project plan using the information gathered here.

Although this research may seem dispensable, it's not. Getting to know the business reasons for your data mining effort helps to ensure that everyone is on the same page before expending valuable resources.

Related information

- [CRISP-DM Help Overview](#)
 - [Determining Business Objectives](#)
 - [Assessing the Situation](#)
 - [Determining Data Mining Goals](#)
 - [Producing a Project Plan](#)
 - [Data Understanding Overview](#)
 - [Data Preparation Overview](#)
 - [Modeling Overview](#)
 - [Evaluation Overview](#)
 - [Deployment Overview](#)
-

Determining Business Objectives

Your first task is to try to gain as much insight as possible into the business goals for data mining. This may not be as easy as it seems, but you can minimize later risk by clarifying problems, goals, and resources.

The CRISP-DM methodology provides a structured way for you to accomplish this.

Task List

- Start gathering [background information](#) about the current business situation.
 - [Document specific business objectives](#) decided upon by key decision makers.
 - Agree upon criteria used to determine [data mining success from a business perspective](#).
 - [E-Retail Example--Finding Business Objectives](#)
 - [Compiling the Business Background](#)
 - [Defining Business Objectives](#)
 - [Business Success Criteria](#)
-

E-Retail Example--Finding Business Objectives

A Web-Mining Scenario Using CRISP-DM

As more companies make the transition to selling over the Web, an established computer/electronics e-retailer is facing increasing competition from newer sites. Faced with the reality that Web stores are cropping up as fast (or faster!) than customers are migrating to the Web, the

company must find ways to remain profitable despite the rising costs of customer acquisition. One proposed solution is to cultivate existing customer relationships in order to maximize the value of each of the company's current customers.

Thus, a study is commissioned with the following objectives:

- Improve cross-sales by making better recommendations.
- Increase customer loyalty with a more personalized service.

Tentatively, the study will be judged a success if:

- Cross-sales increase by 10%.
- Customers spend more time and see more pages on the site per visit.
- The study finishes on time and under budget.

Related information

- [E-Retail Example--Assessing the Situation](#)
 - [Compiling the Business Background](#)
 - [Defining Business Objectives](#)
 - [Business Success Criteria](#)
-

Compiling the Business Background

Understanding your organization's business situation helps you know what you're working with in terms of:

- Available resources (personnel and material)
- Problems
- Goals

You'll need to do a bit of research on the current business situation in order to find real answers to questions that can impact the outcome of the data mining project.

Task 1--Determine Organizational Structure

- Develop organizational charts to illustrate corporate divisions, departments, and project groups. Be sure to include managers' names and responsibilities.
- Identify key individuals in the organization.
- Identify an internal sponsor who will provide financial support and/or domain expertise.
- Determine whether there is a steering committee and procure a list of members.
- Identify business units that will be affected by the data mining project.

Task 2--Describe Problem Area

- Identify the problem area, such as marketing, customer care, or business development.
- Describe the problem in general terms.
- Clarify the prerequisites of the project. What are the motivations behind the project? Does the business already use data mining?
- Check on the status of the data mining project within the business group. Has the effort been approved, or does data mining need to be "advertised" as a key technology for the business group?
- If necessary, prepare informational presentations on data mining to your organization.

Task 3--Describe Current Solution

- Describe any solutions currently used to address the business problem.
- Describe the advantages and disadvantages of the current solution. Also, address the level of acceptance this solution has had within the organization.

Related information

- [E-Retail Example--Finding Business Objectives](#)
 - [Defining Business Objectives](#)
 - [Business Success Criteria](#)
-

Defining Business Objectives

This is where things get specific. As a result of your research and meetings, you should construct a concrete primary objective agreed upon by the project sponsors and other business units affected by the results. This goal will eventually be translated from something as nebulous as "reducing customer churn" to specific data mining objectives that will guide your analytics.

Task List

Be sure to take notes on the following points for later incorporation into the project plan. Remember to keep goals realistic.

- Describe the problem you want to solve using data mining.
- Specify all business questions as precisely as possible.
- Determine any other business requirements (such as not losing any existing customers while increasing cross-sell opportunities).
- Specify expected benefits in business terms (such as reducing churn among high-value customers by 10%).

Related information

- [E-Retail Example--Finding Business Objectives](#)
 - [Compiling the Business Background](#)
 - [Business Success Criteria](#)
-

Business Success Criteria

The goal ahead may be clear, but will you know once you're there? It's important to define the nature of business success for your data mining project before proceeding further. Success criteria fall into two categories:

- **Objective.** These criteria can be as simple as a specific increase in the accuracy of audits or an agreed-upon reduction in churn.
- **Subjective.** Subjective criteria such as "discover clusters of effective treatments" are more difficult to pin down, but you can agree upon who makes the final decision.

Task List

- As precisely as possible, document the success criteria for this project.
- Make sure each business objective has a correlative criterion for success.
- Align the arbiters of the subjective measurements of success. If possible, take notes on their expectations.

Related information

- [E-Retail Example--Finding Business Objectives](#)
 - [Compiling the Business Background](#)
 - [Defining Business Objectives](#)
-

Assessing the Situation

Now that you have a clearly defined goal, it's time to make an assessment of where you are right now. This step involves asking questions such as:

- What sort of data are available for analysis?
 - Do you have the personnel needed to complete the project?
 - What are the biggest risk factors involved?
 - Do you have a contingency plan for each risk?
- [E-Retail Example--Assessing the Situation](#)
 - [Resource Inventory](#)
 - [Requirements, Assumptions, and Constraints](#)
 - [Risks and Contingencies](#)
 - [Terminology](#)
 - [Cost/Benefit Analysis](#)
-

E-Retail Example--Assessing the Situation

A Web-Mining Scenario Using CRISP-DM

This is the electronics e-retailer's first attempt at Web mining, and the company has decided to consult a data mining specialist to help in getting started. One of the first tasks the consultant faces is to assess the company's resources for data mining.

Personnel. It's clear that there is in-house expertise with managing server logs and product and purchase databases, but little experience in data warehousing and data cleaning for analysis. Thus, a database specialist may also be consulted. Since the company hopes the results of the study will become part of a continuing Web-mining process, management must also consider whether any positions created during the current effort will be permanent ones.

Data. Since this is an established company, there is plenty of Web log and purchase data to draw from. In fact, for this initial study, the company will restrict the analysis to customers who have "registered" on the site. If successful, the program can be expanded.

Risks. Aside from the monetary outlays for the consultants and the time spent by employees on the study, there is not a great deal of immediate risk in this venture. However, time is always important, so this initial project is scheduled for a single financial quarter.

Also, there is not a lot of extra cash flow at the moment, so it is imperative that the study come in under budget. If either of these goals should be in danger, the business managers have suggested that the project's scope should be reduced.

Related information

- [Resource Inventory](#)
 - [Requirements, Assumptions, and Constraints](#)
 - [Risks and Contingencies](#)
 - [Terminology](#)
 - [Cost/Benefit Analysis](#)
 - [E-Retail Example--Data Mining Goals](#)
-

Resource Inventory

Taking an accurate inventory of your resources is indispensable. You can save a lot of time and headaches by taking a real look at hardware, data sources, and personnel issues.

Task 1--Research Hardware Resources

- What hardware do you need to support?

Task 2--Identify Data Sources and Knowledge Stores

- Which data sources are available for data mining? Take note of data types and formats.
- How are the data stored? Do you have live access to data warehouses or operational databases?
- Do you plan to purchase external data, such as demographic information?
- Are there any security issues preventing access to required data?

Task 3--Identify Personnel Resources

- Do you have access to business and data experts?
- Have you identified database administrators and other support staff that may be needed?

Once you have asked these questions, include a list of contacts and resources for the phase report.

Related information

- [E-Retail Example--Assessing the Situation](#)
 - [Requirements, Assumptions, and Constraints](#)
 - [Risks and Contingencies](#)
 - [Terminology](#)
 - [Cost/Benefit Analysis](#)
-

Requirements, Assumptions, and Constraints

Your efforts are more likely to pay off if you make an honest assessment of liabilities to the project. Making these concerns as explicit as possible will help to avert future problems.

Task 1--Determine Requirements

The fundamental requirement is the business goal discussed earlier, but consider the following:

- Are there security and legal restrictions on the data or project results?
- Is everyone aligned on the project scheduling requirements?
- Are there requirements on results deployment (for example, publishing to the Web or reading scores into a database)?

Task 2--Clarify Assumptions

- Are there economic factors that might affect the project (for example, consulting fees or competitive products)?
- Are there data quality assumptions?
- How does the project sponsor/management team expect to view the results? In other words, do they want to understand the model itself or simply view the results?

Task 3--Verify Constraints

- Do you have all passwords required for data access?
- Have you verified all legal constraints on data usage?
- Are all financial constraints covered in the project budget?

Related information

- [E-Retail Example--Assessing the Situation](#)
 - [Resource Inventory](#)
 - [Risks and Contingencies](#)
 - [Terminology](#)
 - [Cost/Benefit Analysis](#)
-

Risks and Contingencies

It is also wise to consider possible risks over the course of the project. Types of risks include:

- Scheduling (What if the project takes longer than anticipated?)
- Financial (What if the project sponsor encounters budgetary problems?)
- Data (What if the data are of poor quality or coverage?)
- Results (What if the initial results are less dramatic than expected?)

After you have considered the various risks, come up with a contingency plan to help avert disaster.

Task List

- Document each possible risk.
- Document a contingency plan for each risk.

Related information

- [E-Retail Example--Assessing the Situation](#)
 - [Resource Inventory](#)
 - [Requirements, Assumptions, and Constraints](#)
 - [Terminology](#)
 - [Cost/Benefit Analysis](#)
-

Terminology

To ensure that business and data mining teams are "speaking the same language," you should consider compiling a glossary of technical terms and buzzwords that need clarification. For example, if "churn" for your business has a particular and unique meaning, it is worth explicitly stating that for the benefit of the whole team. Likewise, the team may benefit from clarification of the usage of a gains chart.

Task List

- Keep a list of terms or jargon confusing to team members. Include both business and data mining terminology.
- Consider publishing the list on the intranet or in other project documentation.

Related information

- [E-Retail Example--Assessing the Situation](#)

- [Resource Inventory](#)
 - [Requirements, Assumptions, and Constraints](#)
 - [Risks and Contingencies](#)
 - [Cost/Benefit Analysis](#)
-

Cost/Benefit Analysis

This step answers the question, **What is your bottom line?** As part of the final assessment, it's critical to compare the costs of the project with the potential benefits of success.

Task List

Include in your analysis estimated costs for:

- Data collection and any external data used
- Results deployment
- Operating costs

Then, take into account the benefits of:

- The primary objective being met
- Additional insights generated from data exploration
- Possible benefits from better data understanding

Related information

- [E-Retail Example--Assessing the Situation](#)
 - [Resource Inventory](#)
 - [Requirements, Assumptions, and Constraints](#)
 - [Risks and Contingencies](#)
 - [Terminology](#)
-

Determining Data Mining Goals

Now that the business goal is clear, it's time to translate it into a data mining reality. For example, the business objective to "reduce churn" can be translated into a data mining goal that includes:

- Identifying high-value customers based on recent purchase data
- Building a model using available customer data to predict the likelihood of churn for each customer
- Assigning each customer a rank based on both churn propensity and customer value

These data mining goals, if met, can then be used by the business to reduce churn among the most valuable customers.

As you can see, business and technology must work hand-in-hand for effective data mining. Read on for specific tips on how to determine data mining goals.

- [Data Mining Goals](#)
 - [E-Retail Example--Data Mining Goals](#)
 - [Data Mining Success Criteria](#)
-

Data Mining Goals

As you work with business and data analysts to define a technical solution to the business problem, remember to keep things concrete.

Task List

- Describe the **type** of data mining problem, such as clustering, prediction, or classification.
- Document technical goals using specific units of time, such as predictions with a three-month validity.
- If possible, provide actual numbers for desired outcomes, such as producing churn scores for 80% of existing customers.

Related information

- [E-Retail Example--Data Mining Goals](#)
 - [Data Mining Success Criteria](#)
-

E-Retail Example--Data Mining Goals

A Web-Mining Scenario Using CRISP-DM

With the help of its data mining consultant, the e-retailer has been able to translate the company's business objectives into data mining terms. The goals for the initial study to be completed this quarter are:

- Use historical information about previous purchases to generate a model that links "related" items. When users look at an item description, provide links to other items in the related group (**market basket analysis**).
- Use Web logs to determine what different customers are trying to find, and then redesign the site to highlight these items. Each different customer "type" will see a different main page for the site (**profiling**).
- Use Web logs to try to predict where a person is going next, given where he or she came from and has been on your site (**sequence analysis**).

Related information

- [Data Mining Goals](#)
 - [Data Mining Success Criteria](#)
 - [Sample Project Plan](#)
-

Data Mining Success Criteria

Success must also be defined in technical terms to keep your data mining efforts on track. Use the data mining goal determined earlier to formulate benchmarks for success. IBM® SPSS® Modeler provides tools such as the Evaluation node and the Analysis node to help you analyze the accuracy and validity of your results.

Task List

- Describe the methods for model assessment (for example, accuracy, performance, etc.).
- Define benchmarks for evaluating success. Provide specific numbers.
- Define subjective measurements as best you can and determine the arbiter of success.
- Consider whether the successful deployment of model results is part of data mining success. Start planning now for deployment.

Related information

- [Data Mining Goals](#)
 - [E-Retail Example--Data Mining Goals](#)
-

Producing a Project Plan

At this point, you're ready to produce a plan for the data mining project. The questions you have asked so far and the business and data mining goals you have formulated will form the basis for this road map.

- [Writing the Project Plan](#)
 - [Sample Project Plan](#)
 - [Assessing Tools and Techniques](#)
-

Writing the Project Plan

The project plan is the master document for all of your data mining work. If done well, it can inform everyone associated with the project of the goals, resources, risks, and schedule for all phases of data mining. You may want to publish the plan, as well as documentation gathered throughout this phase, to your company's intranet.

Task List

When creating the plan, be sure you've answered the following questions:

- Have you discussed the project tasks and proposed plan with everyone involved?
- Are time estimates included for all phases or tasks?
- Have you included the effort and resources needed to deploy the results or business solution?
- Are decision points and review requests highlighted in the plan?
- Have you marked phases where multiple iterations typically occur, such as modeling?

Related information

- [Sample Project Plan](#)
 - [Assessing Tools and Techniques](#)
-

Sample Project Plan

The overview plan for the study is as shown in the table below.

Table 1. Sample project plan overview

Phase	Time	Resources	Risks
Business understanding	1 week	All analysts	Economic change
Data understanding	3 weeks	All analysts	Data problems, technology problems
Data preparation	5 weeks	Data mining consultant, some database analyst time	Data problems, technology problems
Modeling	2 weeks	Data mining consultant, some database analyst time	Technology problems, inability to find adequate model
Evaluation	1 week	All analysts	Economic change, inability to implement results
Deployment	1 week	Data mining consultant, some database analyst time	Economic change, inability to implement results

Related information

- [Writing the Project Plan](#)
 - [Assessing Tools and Techniques](#)
-

Assessing Tools and Techniques

Since you've already chosen to use IBM® SPSS® Modeler as your tool for data mining success, you can use this step to research which data mining techniques are most appropriate for your business needs. IBM SPSS Modeler offers a full range of tools for each phase of data mining. To decide when to use the various techniques, consult the modeling section of the online Help.

Related information

- [Writing the Project Plan](#)
 - [Sample Project Plan](#)
 - [Ready for the next step?](#)
-

Ready for the next step?

Before exploring data and beginning work in IBM® SPSS® Modeler, be sure you have answered the following questions.

From a business perspective:

- What does your business hope to gain from this project?
- How will you define the successful completion of our efforts?
- Do you have the budget and resources needed to reach our goals?
- Do you have access to all the data needed for this project?
- Have you and your team discussed the risks and contingencies associated with this project?

- Do the results of your cost/benefit analysis make this project worthwhile?

After you've answered the above questions, did you translate those answers into a data mining goal?

From a data mining perspective:

- How specifically can data mining help you meet your business goals?
- Do you have an idea about which data mining techniques might produce the best results?
- How will you know when your results are accurate or effective enough? (*Have we set a measurement of data mining success?*)
- How will the modeling results be deployed? Have you considered deployment in your project plan?
- Does the project plan include all phases of CRISP-DM?
- Are risks and dependencies called out in the plan?

If you can answer "yes" to the above questions, then you're ready to take a closer look at the data.

Related information

- [Data Understanding Overview](#)
-

Data Understanding

- [Data Understanding Overview](#)
 - [Collecting Initial Data](#)
 - [Describing Data](#)
 - [Exploring Data](#)
 - [Verifying Data Quality](#)
 - [Ready for the next step?](#)
-

Data Understanding Overview

The data understanding phase of CRISP-DM involves taking a closer look at the data available for mining. This step is critical in avoiding unexpected problems during the next phase--data preparation--which is typically the longest part of a project.

Data understanding involves accessing the data and exploring it using tables and graphics that can be organized in IBM® SPSS® Modeler using the CRISP-DM project tool. This enables you to determine the quality of the data and describe the results of these steps in the project documentation.

Related information

- [CRISP-DM Help Overview](#)
 - [Collecting Initial Data](#)
 - [Describing Data](#)
 - [Exploring Data](#)
 - [Verifying Data Quality](#)
 - [Business Understanding Overview](#)
 - [Data Preparation Overview](#)
 - [Modeling Overview](#)
 - [Evaluation Overview](#)
 - [Deployment Overview](#)
-

Collecting Initial Data

At this point in CRISP-DM, you're ready to access data and bring it into IBM® SPSS® Modeler. Data come from a variety of sources, such as:

- **Existing data.** This includes a wide variety of data, such as transactional data, survey data, Web logs, etc. Consider whether the existing data are enough to meet your needs.
- **Purchased data.** Does your organization use supplemental data, such as demographics? If not, consider whether it may be needed.
- **Additional data.** If the above sources don't meet your needs, you may need to conduct surveys or begin additional tracking to supplement the existing data stores.

Task List

Take a look at the data in IBM SPSS Modeler and consider the following questions. Be sure to take notes on your findings. See the topic [Writing a Data Collection Report](#) for more information.

- Which attributes (columns) from the database seem most promising?
 - Which attributes seem irrelevant and can be excluded?
 - Is there enough data to draw generalizable conclusions or make accurate predictions?
 - Are there too many attributes for your modeling method of choice?
 - Are you merging various data sources? If so, are there areas that might pose a problem when merging?
 - Have you considered how missing values are handled in each of your data sources?
- [E-Retail Example--Initial Data Collection](#)
• [Writing a Data Collection Report](#)
-

E-Retail Example--Initial Data Collection

A Web-Mining Scenario Using CRISP-DM

The e-retailer in this example uses several important data sources, including:

Web logs. The raw access logs contain all of the information on how customers navigate the Web site. References to image files and other non-informative entries in the Web logs will need to be removed as part of the data preparation process.

Purchase data. When a customer submits an order, all of the information pertinent to that order is saved. The orders in the purchase database need to be mapped to the corresponding sessions in the Web logs.

Product database. The product attributes may be useful when determining "related" products. The product information needs to be mapped to the corresponding orders.

Customer database. This database contains extra information collected from registered customers. The records are by no means complete, because many customers do not fill out questionnaires. The customer information needs to be mapped to the corresponding purchases and sessions in the Web logs.

At this moment, the company has no plans to purchase external databases or spend money conducting surveys because its analysts are busy managing the data they currently have. At some point, however, they may want to consider an extended deployment of data mining results, in which case purchasing additional demographic data for unregistered customers may be quite useful. It may also be useful to have demographic information to see how the e-retailer's customer base differs from the average Web shopper.

Related information

- [Writing a Data Collection Report](#)
-

Writing a Data Collection Report

Using the material gathered in the previous step, you can begin to write a data collection report. Once complete, the report can be added to the project Web site or distributed to the team. It can also be combined with the reports prepared in the next steps--data description, exploration, and quality verification. These reports will guide your work throughout the data preparation phase.

Related information

- [E-Retail Example--Initial Data Collection](#)
-

Describing Data

There are many ways to describe data, but most descriptions focus on the quantity and quality of the data--how much data is available and the condition of the data. Listed below are some key characteristics to address when describing data.

- **Amount of data.** For most modeling techniques, there are trade-offs associated with data size. Large data sets can produce more accurate models, but they can also lengthen the processing time. Consider whether using a subset of data is a possibility. When taking notes for the final report, be sure to include size statistics for all data sets, and remember to consider both the number of records as well as fields (attributes) when describing data.

- **Value types.** Data can take a variety of formats, such as **numeric**, **categorical** (string), or **Boolean** (true/false). Paying attention to value type can head off problems during later modeling.
- **Coding schemes.** Frequently, values in the database are representations of characteristics such as gender or product type. For example, one data set may use *M* and *F* to represent *male* and *female*, while another may use the numeric values 1 and 2. Note any conflicting schemes in the data report.

With this knowledge in hand, you are now ready to write the [data description report](#) and share your findings with a larger audience.

- [E-Retail Example--Describing Data](#)
 - [Writing a Data Description Report](#)
-

E-Retail Example--Describing Data

A Web-Mining Scenario Using CRISP-DM

There are many records and attributes to process in a Web-mining application. Even though the e-retailer conducting this data mining project has limited the initial study to the approximately 30,000 customers who have registered on the site, there are still millions of records in the Web logs.

Most of the value types in these data sources are symbolic, whether they are dates and times, Web pages accessed, or answers to multiple-choice questions from the registration questionnaire. Some of these variables will be used to create new variables that are numeric, such as number of Web pages visited and time spent at the Web site. The few existing numeric variables in the data sources include the number of each product ordered, the amount spent during a purchase, and product weight and dimension specifications from the product database.

There is little overlap in the coding schemes for the various data sources because the data sources contain very different attributes. The only variables that overlap are "keys," such as the customer IDs and product codes. These variables must have identical coding schemes from data source to data source; otherwise, it would be impossible to merge the data sources. Some additional data preparation will be necessary to recode these key fields for merging.

Related information

- [Writing a Data Description Report](#)
-

Writing a Data Description Report

To proceed effectively with your data mining project, consider the value of producing an accurate data description report using the following metrics:

Data Quantity

- What is the format of the data?
- Identify the method used to capture the data--for example, ODBC.
- How large is the database (in numbers of rows and columns)?

Data Quality

- Does the data include characteristics relevant to the business question?
- What data types are present (symbolic, numeric, etc.)?
- Did you compute basic statistics for the key attributes? What insight did this provide into the business question?
- Are you able to prioritize relevant attributes? If not, are business analysts available to provide further insight?

Related information

- [E-Retail Example--Describing Data](#)
-

Exploring Data

Use this phase of CRISP-DM to explore the data with the tables, charts, and other visualization tools available in IBM® SPSS® Modeler. Such analyses can help to address the data mining goal constructed during the [business understanding](#) phase. They can also help to formulate hypotheses and shape the data transformation tasks that take place during data preparation.

- [E-Retail Example--Exploring Data](#)

- [Writing a Data Exploration Report](#)
-

E-Retail Example--Exploring Data

A Web-Mining Scenario Using CRISP-DM

Although CRISP-DM suggests conducting an initial exploration at this point, data exploration is difficult, if not impossible, on raw Web logs, as our e-retailer has found out. Typically, Web log data must be processed first in the data preparation phase to produce data that can be meaningfully explored. This departure from CRISP-DM underscores the fact that the process can and should be customized for your particular data mining needs. CRISP-DM is cyclical, and data miners typically move back and forth between phases.

Although Web logs must be processed before exploration, the other data sources available to the e-retailer are more amenable to exploration. Using the purchase database for exploration reveals interesting summaries about customers, such as how much they spend, how many items they buy per purchase, and where they come from. Summaries of the customer database will show the distribution of responses to the items on the registration questionnaire.

Exploration is also useful for looking for errors in the data. While most of the data sources are automatically generated, information in the product database was entered by hand. Some quick summaries of listed product dimensions will help to discover typos, such as "119-inch" (instead of "19-inch") monitor.

Related information

- [Writing a Data Exploration Report](#)
-

Writing a Data Exploration Report

As you create graphs and run statistics on the available data, start forming hypotheses about how the data can answer the technical and business goals.

Task List

Take notes on your findings for inclusion in the data exploration report. Be sure to answer the following questions:

- What sort of hypotheses have you formed about the data?
- Which attributes seem promising for further analysis?
- Have your explorations revealed new characteristics about the data?
- How have these explorations changed your initial hypothesis?
- Can you identify particular subsets of data for later use?
- Take another look at your data mining goals. Has this exploration altered the goals?

Related information

- [E-Retail Example--Exploring Data](#)
-

Verifying Data Quality

Data are rarely perfect. In fact, most data contain coding errors, missing values, or other types of inconsistencies that make analysis tricky at times. One way to avoid potential pitfalls is to conduct a thorough quality analysis of available data before modeling.

The reporting tools in IBM® SPSS® Modeler (such as the Data Audit, Table and other output nodes) can help you look for the following types of problems:

- **Missing data** include values that are blank or coded as a non-response (such as \$null\$, ?, or 999).
- **Data errors** are usually typographical errors made in entering the data.
- **Measurement errors** include data that are entered correctly but are based on an incorrect measurement scheme.
- **Coding inconsistencies** typically involve nonstandard units of measurement or value inconsistencies, such as the use of both *M* and *male* for gender.
- **Bad metadata** include mismatches between the apparent meaning of a field and the meaning stated in a field name or definition.

Be sure to take notes on such quality concerns. See the topic [Writing a Data Quality Report](#) for more information.

- [E-Retail Example--Verifying Data Quality](#)
 - [Writing a Data Quality Report](#)
-

E-Retail Example--Verifying Data Quality

A Web-Mining Scenario Using CRISP-DM

The verification of data quality is often accomplished during the course of the description and exploration processes. Some of the issues encountered by the e-retailer include:

Missing Data. The known missing data includes the unanswered questionnaires by some of the registered users. Without the extra information provided by the questionnaire, these customers may have to be left out of some of the subsequent models.

Data Errors. Most of the data sources are automatically generated, so this is not a great worry. Typographical errors in the product database can be found during the exploration process.

Measurement Errors. The greatest potential source for measurement error is the questionnaire. If any of the items are ill-advised or poorly worded, they may not provide the information the e-retailer hopes to obtain. Again, during the exploration process, it is important to pay special attention to items that have an unusual distribution of answers.

Related information

- [Writing a Data Quality Report](#)
-

Writing a Data Quality Report

Based on your exploration and verification of data quality, you're now ready to prepare a report that will guide the next phase of CRISP-DM. See the topic [Verifying Data Quality](#) for more information.

Task List

As discussed earlier, there are [several types of data quality problems](#). Before moving to the next step, consider the following quality concerns and plan for a solution. Document all responses in the data quality report.

- Have you identified missing attributes and blank fields? If so, is there meaning behind such missing values?
- Are there spelling inconsistencies that may cause problems in later merges or transformations?
- Have you explored deviations to determine whether they are "noise" or phenomena worth analyzing further?
- Have you conducted a plausibility check for values? Take notes on any apparent conflicts (such as teenagers with high income levels).
- Have you considered excluding data that has no impact on your hypotheses?
- Are the data stored in flat files? If so, are the delimiters consistent among files? Does each record contain the same number of fields?

Related information

- [E-Retail Example--Verifying Data Quality](#)
 - [Ready for the next step?](#)
-

Ready for the next step?

Before preparing the data for modeling in IBM® SPSS® Modeler, consider the following points:

How well do you understand the data?

- Are all data sources clearly identified and accessed? Are you aware of any problems or restrictions?
- Have you identified key attributes from the available data?
- Did these attributes help you to formulate hypotheses?
- Have you noted the size of all data sources?
- Are you able to use a subset of data where appropriate?
- Have you computed basic statistics for each attribute of interest? Did meaningful information emerge?
- Did you use exploratory graphics to gain further insight into key attributes? Did this insight reshape any of your hypotheses?
- What are the data quality issues for this project? Do you have a plan to address these issues?
- Are the data preparation steps clear? For instance, do you know which data sources to merge and which attributes to filter or select?

Now that you're armed with both business and data understanding, it's time to use IBM SPSS Modeler to prepare your data for modeling.

Related information

- [Data Preparation Overview](#)
-

Data Preparation

- [Data Preparation Overview](#)
 - [Selecting Data](#)
 - [Cleaning Data](#)
 - [Constructing New Data](#)
 - [Integrating Data](#)
 - [Formatting Data](#)
 - [Ready for modeling?](#)
-

Data Preparation Overview

Data preparation is one of the most important and often time-consuming aspects of data mining. In fact, it is estimated that data preparation usually takes 50-70% of a project's time and effort. Devoting adequate energy to the earlier [business understanding](#) and [data understanding](#) phases can minimize this overhead, but you still need to expend a good amount of effort preparing and packaging the data for mining.

Depending on your organization and its goals, data preparation typically involves the following tasks:

- Merging data sets and/or records
- Selecting a sample subset of data
- Aggregating records
- Deriving new attributes
- Sorting the data for modeling
- Removing or replacing blank or missing values
- Splitting into training and test data sets

Related information

- [CRISP-DM Help Overview](#)
 - [Selecting Data](#)
 - [Cleaning Data](#)
 - [Constructing New Data](#)
 - [Integrating Data](#)
 - [Formatting Data](#)
 - [Business Understanding Overview](#)
 - [Data Understanding Overview](#)
 - [Modeling Overview](#)
 - [Evaluation Overview](#)
 - [Deployment Overview](#)
-

Selecting Data

Based upon the [initial data collection](#) conducted in the previous CRISP-DM phase, you are ready to begin selecting the data relevant to your data mining goals. Generally, there are two ways to select data:

- **Selecting items (rows)** involves making decisions such as which accounts, products, or customers to include.
- **Selecting attributes or characteristics (columns)** involves making decisions about the use of characteristics such as transaction amount or household income.
- [E-Retail Example--Selecting Data](#)
- [Including or Excluding Data](#)

E-Retail Example--Selecting Data

A Web-Mining Scenario Using CRISP-DM

Many of the e-retailer's decisions about which data to select have already been made in earlier phases of the data mining process.

Selecting items. The initial study will be limited to the (approximately) 30,000 customers who have registered on the site, so filters need to be set up to exclude purchases and Web logs of nonregistered customers. Other filters should be established to remove calls to image files and other non-informative entries in the Web logs.

Selecting attributes. The purchase database will contain sensitive information about the e-retailer's customers, so it is important to filter attributes such as the customer name, address, phone number, and credit card numbers.

Related information

- [Including or Excluding Data](#)

Including or Excluding Data

As you decide upon subsets of data to include or exclude, be sure to document the rationale behind your decisions.

Questions to Consider

- Is a given attribute relevant to your data mining goals?
- Does the quality of a particular data set or attribute preclude the validity of your results?
- Can you salvage such data?
- Are there any constraints on using particular fields such as *gender* or *race*?

Are your decisions here different than the hypotheses formulated in the data understanding phase? If so, be sure to document your reasoning in the project report.

Related information

- [E-Retail Example--Selecting Data](#)

Cleaning Data

Cleaning your data involves taking a closer look at the problems in the data that you've chosen to include for analysis. There are several ways to clean data using the Record and Field Operation nodes in IBM® SPSS® Modeler.

Table 1. Cleaning data

Data Problem	Possible Solution
Missing data	Exclude rows or characteristics. Or, fill blanks with an estimated value.
Data errors	Use logic to manually discover errors and replace. Or, exclude characteristics.
Coding inconsistencies	Decide upon a single coding scheme, then convert and replace values.
Missing or bad metadata	Manually examine suspect fields and track down correct meaning.

The [Data Quality Report](#) prepared during the data understanding phase contains details about the types of problems particular to your data. You can use it as a starting point for data manipulation in IBM SPSS Modeler.

- [E-Retail Example--Cleaning Data](#)
- [Writing a Data Cleaning Report](#)

Related information

- [E-Retail Example--Constructing Data](#)
- [E-Retail Example--Integrating Data](#)

E-Retail Example--Cleaning Data

The e-retailer uses the data cleaning process to address the problems noted in the data quality report.

Missing data. Customers who did not complete the online questionnaire may have to be left out of some of the models later on. These customers could be asked again to fill out the questionnaire, but this will take time and money that the e-retailer cannot afford to spend. What the e-retailer can do is model the purchasing differences between customers who do and do not answer the questionnaire. If these two sets of customers have similar purchasing habits, the missing questionnaires are less worrisome.

Data errors. Errors found during the exploration process can be corrected here. For the most part, though, proper data entry is enforced on the Web site before a customer submits a page to the back-end database.

Measurement errors. Poorly worded items on the questionnaire can greatly affect the quality of the data. As with missing questionnaires, this is a difficult problem because there may not be time or money available to collect answers to a new replacement question. For problematic items, the best solution may be to go back to the selection process and filter these items from further analyses.

Related information

- [Writing a Data Cleaning Report](#)
 - [E-Retail Example--Constructing Data](#)
 - [E-Retail Example--Integrating Data](#)
-

Writing a Data Cleaning Report

Reporting your data-cleaning efforts is essential for tracking alterations to the data. Future data mining projects will benefit from having the details of your work readily available.

Task List

It's a good idea to consider the following questions when writing the report:

- What types of noise occurred in the data?
- What approaches did you use to remove the noise? Which techniques were successful?
- Are there any cases or attributes that could not be salvaged? Be sure to note data excluded due to noise.

Related information

- [E-Retail Example--Cleaning Data](#)
 - [E-Retail Example--Constructing Data](#)
 - [E-Retail Example--Integrating Data](#)
-

Constructing New Data

It is frequently the case that you'll need to construct new data. For example, it may be useful to create a new column flagging the purchase of an extended warranty for each transaction. This new field, *purchased_warranty*, can easily be generated using a Set to Flag node in IBM® SPSS® Modeler.

There are two ways to construct new data:

- Deriving attributes (columns or characteristics)
- Generating records (rows)

IBM SPSS Modeler offers a multitude of ways to construct data using its Record and Field Operations nodes.

- [E-Retail Example--Constructing Data](#)
 - [Deriving Attributes](#)
-

E-Retail Example--Constructing Data

The processing of Web logs can create many new attributes. For the events recorded in the logs, the e-retailer will want to create timestamps, identify visitors and sessions, and note the page accessed and the type of activity the event represents. Some of these variables will be used to create more attributes, such as the time between events within a session.

Further attributes can be created as a result of a merge or other data restructuring. For example, when the event-per-row Web logs are "rolled up" so that each row is a session, new attributes recording the total number of actions, total time spent, and total purchases made during the session will be created. When the Web logs are merged with the customer database so that each row is a customer, new attributes recording the number of sessions, total number of actions, total time spent, and total purchases made by each customer will be created.

After constructing new data, the e-retailer goes through an exploration process to make sure that the data creation was performed correctly.

Related information

- [Cleaning Data](#)
 - [E-Retail Example--Cleaning Data](#)
 - [Writing a Data Cleaning Report](#)
 - [E-Retail Example--Integrating Data](#)
-

Deriving Attributes

In IBM® SPSS® Modeler, you can use the following Field Operations nodes to derive new attributes:

- Create new fields derived from existing ones using a **Derive node**.
- Create a flag field using a **Set to Flag node**.

Task List

- Consider the data requirements for modeling when deriving attributes. Does the modeling algorithm expect a particular type of data, such as numeric? If so, perform the necessary transformations.
 - Do the data need be normalized before modeling?
 - Can missing attributes be constructed using aggregation, averaging, or induction?
 - Based upon your background knowledge, are there important facts (such as length of time spent at the Web site) that can be derived from existing fields?
-

Integrating Data

It is not uncommon to have multiple data sources for the same set of business questions. For example, you may have access to mortgage loan data as well as purchased demographic data for the same set of clients. If these data sets contain the same unique identifier (such as social security number), you can merge them in IBM® SPSS® Modeler using this key field.

There are two basic methods of integrating data:

- **Merging** data involves merging two data sets with similar records but different attributes. The data is merged using the same key identifier for each record (such as customer ID). The resulting data increases in columns or characteristics.
 - **Appending** data involves integrating two or more data sets with similar attributes but different records. The data is integrated based upon a similar fields (such as product name or contract length).
- [E-Retail Example--Integrating Data](#)
 - [Integration Tasks](#)
-

E-Retail Example--Integrating Data

A Web-Mining Scenario Using CRISP-DM

With multiple data sources, there are many different ways in which the e-retailer can integrate data:

- **Adding customer and product attributes to event data.** In order to model Web log events using attributes from other databases, any customer ID, product number, and purchase order number associated with each event must be correctly identified and the corresponding attributes merged to the processed Web logs. Note that the merged file replicates customer and product information every time a customer or product is associated with an event.
- **Adding purchase and Web log information to customer data.** In order to model the value of a customer, their purchases and session information must be picked out of the appropriate databases, totaled, and merged with the customer database. This involves the creation

of new attributes as discussed in the constructing data process.

After integrating databases, the e-retailer goes through an exploration process to make sure that the data merge was performed correctly.

Related information

- [Cleaning Data](#)
 - [E-Retail Example--Cleaning Data](#)
 - [Writing a Data Cleaning Report](#)
 - [E-Retail Example--Constructing Data](#)
-

Integration Tasks

Integrating data can become complex if you have not spent adequate time developing an understanding of your data. Give some thought to items and attributes that seem most relevant to the data mining goals and then get started integrating your data.

Task List

- Using Merge or Append nodes in IBM® SPSS® Modeler, integrate the data sets considered useful for modeling.
 - Consider saving the resulting output before proceeding to modeling.
 - After merging, data can be simplified by **aggregating** values. Aggregation means that new values are computed by summarizing information from multiple records and/or tables.
 - You may also need to generate new records (such as the average deduction from several years of combined tax returns).
-

Formatting Data

As a final step before model building, it is helpful to check whether certain techniques require a particular format or order to the data. For example, it is not uncommon that a sequence algorithm requires the data to be presorted before running the model. Even if the model can perform the sorting for you, it may save processing time to use a Sort node prior to modeling.

Task List

Consider the following questions when formatting data:

- Which models do you plan to use?
- Do these models require a particular data format or order?

If changes are recommended, the processing tools in IBM® SPSS® Modeler can help you apply the necessary data manipulation.

Ready for modeling?

Before building models in IBM® SPSS® Modeler, be sure you have answered the following questions.

- Are all the data accessible from within IBM SPSS Modeler?
- Based upon your initial exploration and understanding, were you able to select relevant subsets of data?
- Have you cleaned the data effectively or removed unsalvageable items? Document any decisions in the final report.
- Are multiple data sets integrated properly? Were there any merging problems that should be documented?
- Have you researched the requirements of the modeling tools that you plan to use?
- Are there any formatting issues you can address before modeling? This includes both required formatting concerns as well as tasks that may reduce modeling time.

If you can answer the above questions, then you're ready for the crux of data mining--modeling.

Related information

- [Modeling Overview](#)
-

Modeling

- [Modeling Overview](#)
 - [Selecting Modeling Techniques](#)
 - [Generating a Test Design](#)
 - [Building the Models](#)
 - [Assessing the Model](#)
 - [Ready for the next step?](#)
-

Modeling Overview

This is the point at which your hard work begins to pay off. The data you spent time preparing are brought into the analysis tools in IBM® SPSS® Modeler, and the results begin to shed some light on the business problem posed during [Business Understanding](#).

Modeling is usually conducted in multiple iterations. Typically, data miners run several models using the default parameters and then fine-tune the parameters or revert to the data preparation phase for manipulations required by their model of choice. It is rare for an organization's data mining question to be answered satisfactorily with a single model and a single execution. This is what makes data mining so interesting--there are many ways to look at a given problem, and IBM SPSS Modeler offers a wide variety of tools to help you do so.

Related information

- [CRISP-DM Help Overview](#)
 - [Selecting Modeling Techniques](#)
 - [Generating a Test Design](#)
 - [Building the Models](#)
 - [Assessing the Model](#)
 - [Business Understanding Overview](#)
 - [Data Understanding Overview](#)
 - [Data Preparation Overview](#)
 - [Evaluation Overview](#)
 - [Deployment Overview](#)
-

Selecting Modeling Techniques

Although you may already have some idea about which types of modeling are most appropriate for your organization's needs, now is the time to make some firm decisions about which ones to use. Determining the most appropriate model will typically be based on the following considerations:

- **The data types available for mining.** For example, are the fields of interest categorical (symbolic)?
- **Your data mining goals.** Do you simply want to gain insight into transactional data stores and unearth interesting purchase patterns? Or do you need to produce a score indicating, for example, propensity to default on a student loan?
- **Specific modeling requirements.** Does the model require a particular data size or type? Do you need a model with easily presentable results?

For more information on the model types in IBM® SPSS® Modeler and their requirements, see the IBM SPSS Modeler documentation or online Help.

- [E-Retail Example--Modeling Techniques](#)
 - [Choosing the Right Modeling Techniques](#)
 - [Modeling Assumptions](#)
-

E-Retail Example--Modeling Techniques

The modeling techniques employed by the e-retailer are driven by the company's data mining goals:

Improved recommendations. At its simplest, this involves clustering purchase orders to determine which products are most often bought together. Customer data, and even visit records, can be added for richer results. The two-step or Kohonen network clustering techniques are suited for this type of modeling. Afterward, the clusters can be profiled using a C5.0 ruleset to determine which recommendations are most appropriate at any point during a customer's visit.

Improved site navigation. For now, the e-retailer will focus on identifying pages that are often used but require several clicks for the user to find. This entails applying a sequencing algorithm to the Web logs in order to generate the "unique paths" customers take through the Web site, and then specifically looking for sessions that have a lot of page visits without (or before) an action taken. Later, in a more in-depth analysis,

clustering techniques can be used to identify different "types" of visits and visitors, and the site content can be organized and presented according to type.

Related information

- [Choosing the Right Modeling Techniques](#)
 - [Modeling Assumptions](#)
-

Choosing the Right Modeling Techniques

Many modeling techniques are available in IBM® SPSS® Modeler. Frequently, data miners use more than one to approach the problem from a number of directions.

Task List

When deciding on which model(s) to use, consider whether the following issues have an impact on your choices:

- Does the model require the data to be split into test and training sets?
- Do you have enough data to produce reliable results for a given model?
- Does the model require a certain level of data quality? Can you meet this level with the current data?
- Are your data the proper type for a particular model? If not, can you make the necessary conversions using data manipulation nodes?

For more information on the model types in IBM SPSS Modeler and their requirements, see the IBM SPSS Modeler documentation or online Help.

Related information

- [E-Retail Example--Modeling Techniques](#)
 - [Modeling Assumptions](#)
-

Modeling Assumptions

As you begin to narrow down your modeling tools of choice, take notes on the decision-making process. Document any data assumptions as well as any data manipulations made to meet the model's requirements.

For example, both the Logistic Regression and Neural Net nodes require the data types to be fully **instantiated** (data types are known) before execution. This means you will need to add a Type node to the stream and execute it to run the data through before building and running a model. Similarly, predictive models, such as C5.0, may benefit from rebalancing the data when predicting rules for rare events. When making this type of prediction, you can often get better results by inserting a Balance node into the stream and feeding the more balanced subset into the model.

Be sure to document these types of decisions.

Related information

- [E-Retail Example--Modeling Techniques](#)
 - [Choosing the Right Modeling Techniques](#)
-

Generating a Test Design

As a final step before actually building the model, you should take a moment to consider again how the model's results will be tested. There are two parts to generating a comprehensive test design:

- Describing the criteria for "goodness" of a model
- Defining the data on which these criteria will be tested

A model's **goodness** can be measured in several ways. For supervised models, such as C5.0 and C&R Tree, measurements of goodness typically estimate the error rate of a particular model. For unsupervised models, such as Kohonen cluster nets, measurements may include criteria such as ease of interpretation, deployment, or required processing time.

Remember, model building is an iterative process. This means that you will typically test the results of several models before deciding on the ones to use and deploy.

- [Writing a Test Design](#)
 - [E-Retail Example--Test Design](#)
-

Writing a Test Design

The test design is a description of the steps you will take to test the models produced. Because modeling is an iterative process, it is important to know when to stop adjusting parameters and try another method or model.

Task List

When creating a test design, consider the following questions:

- What data will be used to test the models? Have you partitioned the data into train/test sets? (This is a commonly used approach in modeling.)
- How might you measure the success of supervised models (such as C5.0)?
- How might you measure the success of unsupervised models (such as Kohonen cluster nets)?
- How many times are you willing to rerun a model with adjusted settings before attempting another type of model?

Related information

- [E-Retail Example--Test Design](#)
-

E-Retail Example--Test Design

A Web-Mining Scenario Using CRISP-DM

The criteria by which the models are assessed depend on the models under consideration and the data mining goals:

Improved recommendations. Until the improved recommendations are presented to live customers, there is no purely objective way to assess them. However, the e-retailer may require the rules that generate the recommendations to be simple enough to make sense from a business perspective. Likewise, the rules should be complex enough to generate different recommendations for different customers and sessions.

Improved site navigation. Given the evidence of what pages customers access on the Web site, the e-retailer can objectively assess the updated site design in terms of ease of access to important pages. However, as with the recommendations, it is difficult to assess in advance how well customers will adjust to the reorganized site. If time and finances allow, some usability testing may be in order.

Related information

- [Writing a Test Design](#)
-

Building the Models

At this point, you should be well prepared to build the models you've spent so long considering. Give yourself time and room to experiment with a number of different models before making final conclusions. Most data miners typically build several models and compare the results before deploying or integrating them.

In order to track your progress with a variety of models, be sure to keep notes on the settings and data used for each model. This will help you to discuss the results with others and retrace your steps if necessary. At the end of the model-building process, you'll have three pieces of information to use in data mining decisions:

- **Parameter settings** include the notes you take on parameters that produce the best results.
 - The actual **models** produced.
 - **Descriptions of model results**, including performance and data issues that occurred during the execution of the model and exploration of its results.
- [E-Retail Example--Model Building](#)
 - [Parameter Settings](#)
 - [Running the Models](#)
 - [Model description](#)

E-Retail Example--Model Building

A Web-Mining Scenario Using CRISP-DM

Improved recommendations. Clusterings are produced for varying levels of data integration, starting with just the purchase database and then including related customer and session information. For each level of integration, clusterings are produced under varying parameter settings for the two-step and Kohonen network algorithms. For each of these clusterings, a few C5.0 rulesets are generated with different parameter settings.

Improved site navigation. The Sequence modeling node is used to generate customer paths. The algorithm allows the specification of a minimum support criterion, which is useful for focusing on the most common customer paths. Various settings for the parameters are tried.

Related information

- [Parameter Settings](#)
- [Running the Models](#)
- [Model description](#)

Parameter Settings

Most modeling techniques have a variety of parameters or settings that can be adjusted to control the modeling process. For example, decision trees can be controlled by adjusting tree depth, splits, and a number of other settings. Typically, most people build a model first using the default options and then refine parameters during subsequent sessions.

Once you have determined the parameters that produce the most accurate results, be sure to save the stream and generated model nodes. Also, taking notes on the optimal settings can help when you decide to automate or rebuild the model with new data.

Related information

- [E-Retail Example--Model Building](#)
- [Running the Models](#)
- [Model description](#)

Running the Models

In IBM® SPSS® Modeler, running models is a straightforward task. Once you've inserted the model node into the stream and edited any parameters, simply execute the model to produce viewable results. Results appear in the Generated Models navigator on the right side of the workspace. You can right-click a model to browse the results. For most models, you can insert the generated model into the stream to further evaluate and deploy the results. Models can be also be saved in IBM SPSS Modeler for easy reuse.

Related information

- [E-Retail Example--Model Building](#)
- [Parameter Settings](#)
- [Model description](#)

Model description

When examining the results of a model, be sure to take notes on your modeling experience. You can store notes with the model itself using the node annotations dialog box or the project tool.

Task list

For each model, record information such as:

- Can you draw meaningful conclusions from this model?

- Are there new insights or unusual patterns revealed by the model?
- Were there execution problems for the model? How reasonable was the processing time?
- Did the model have difficulties with data quality issues, such as a high number of missing values?
- Were there any calculation inconsistencies that should be noted?

Related information

- [E-Retail Example--Model Building](#)
 - [Parameter Settings](#)
 - [Running the Models](#)
-

Assessing the Model

Now that you have a set of initial models, take a closer look at them to determine which are accurate or effective enough to be final. Final can mean several things, such as "ready to deploy" or "illustrating interesting patterns." Consulting the test plan that you created earlier can help to make this assessment from your organization's point of view.

- [Comprehensive Model Assessment](#)
 - [E-Retail Example--Model Assessment](#)
 - [Keeping Track of Revised Parameters](#)
-

Comprehensive Model Assessment

For each model under consideration, it is a good idea to make a methodical assessment based on the criteria generated in your test plan. Here is where you may add the generated model to the stream and use evaluation charts or analysis nodes to analyze the effectiveness of the results. You should also consider whether the results make logical sense or whether they are too simplistic for your business goals (for example, a sequence that reveals purchases such as wine > wine > wine).

Once you've made an assessment, rank the models in order based on both objective (model accuracy) and subjective (ease of use or interpretation of results) criteria.

Task List

- Using the data mining tools in IBM® SPSS® Modeler, such as evaluation charts, analysis nodes, or cross-validation charts, evaluate the results of your model.
- Conduct a review of the results based on your understanding of the business problem. Consult data analysts or other experts who may have insight into the relevance of particular results.
- Consider whether a model's results are easily deployable. Does your organization require that results be deployed over the Web or sent back to the data warehouse?
- Analyze the impact of results on your success criteria. Do they meet the goals established during the business understanding phase?

If you were able to address the above issues successfully and believe that the current models meet your goals, it's time to move on to a more thorough evaluation of the models and a final deployment. Otherwise, take what you've learned and rerun the models with adjusted parameter settings.

Related information

- [E-Retail Example--Model Assessment](#)
 - [Keeping Track of Revised Parameters](#)
 - [Ready for the next step?](#)
-

E-Retail Example--Model Assessment

A Web-Mining Scenario Using CRISP-DM

Improved recommendations. One of the Kohonen networks and a two-step clustering each give reasonable results, and the e-retailer finds it difficult to choose between them. In time, the company hopes to use both, accepting the recommendations that the two techniques agree on and studying in greater detail the situations in which they differ. With a little effort and applied business knowledge, the e-retailer can develop further rules to resolve differences between the two techniques.

The e-retailer also finds that the results that include the session information are surprisingly good. There is evidence to suggest that recommendations could be tied to site navigation. A ruleset, defining where the customer is likely to go next, could be used in real time to affect the site content directly as the customer is browsing.

Improved site navigation. The Sequence model provides the e-retailer with a high level of confidence that certain customer paths can be predicted, producing results that suggest a manageable number of changes to the site design.

Related information

- [Comprehensive Model Assessment](#)
 - [Keeping Track of Revised Parameters](#)
-

Keeping Track of Revised Parameters

Based on what you've learned during model assessment, it's time to have another look at the models. You have two options here:

- Adjust the parameters of existing models.
- Choose a different model to address your data mining problem.

In both cases, you'll be returning to the building models task and iterate until the results are successful. Don't worry about repeating this step. It is extremely common for data miners to evaluate and rerun models several times before finding one that meets their needs. This is a good argument for building several models at once and comparing the results before adjusting the parameters for each.

Related information

- [Comprehensive Model Assessment](#)
 - [E-Retail Example--Model Assessment](#)
 - [Building the Models](#)
 - [Ready for the next step?](#)
-

Ready for the next step?

Before moving on to a final evaluation of the models, consider whether your initial assessment was thorough enough.

Task List

- Are you able to understand the results of the models?
- Do the model results make sense to you from a purely logical perspective? Are there apparent inconsistencies that need further exploration?
- From your initial glance, do the results seem to address your organization's business question?
- Have you used analysis nodes and lift or gains charts to compare and evaluate model accuracy?
- Have you explored more than one type of model and compared the results?
- Are the results of your model deployable?

If the results of your data modeling seem accurate and relevant, it's time to conduct a more thorough evaluation before a final deployment.

Related information

- [Evaluation Overview](#)
-

Evaluation

- [Evaluation Overview](#)
 - [Evaluating the Results](#)
 - [Review Process](#)
 - [Determining the Next Steps](#)
-

Evaluation Overview

At this point, you've completed most of your data mining project. You've also determined, in the Modeling phase, that the models built are technically correct and effective according to the **data mining success criteria** that you defined earlier.

Before continuing, however, you should evaluate the results of your efforts using the **business success criteria** established at the beginning of the project. This is the key to ensuring that your organization can make use of the results you've obtained. Two types of results are produced by data mining:

- The final **models** selected in the previous phase of CRISP-DM.
- Any conclusions or inferences drawn from the models themselves as well as from the data mining process. These are known as **findings**.

Related information

- [CRISP-DM Help Overview](#)
 - [Evaluating the Results](#)
 - [Review Process](#)
 - [Determining the Next Steps](#)
 - [Business Understanding Overview](#)
 - [Data Understanding Overview](#)
 - [Data Preparation Overview](#)
 - [Modeling Overview](#)
 - [Deployment Overview](#)
-

Evaluating the Results

At this stage, you formalize your assessment of whether or not the project results meet the business success criteria. This step requires a clear understanding of the stated business goals, so be sure to include key decision makers in the project assessment.

Task List

First, you need to document your assessment of whether the data mining results meet the business success criteria. Consider the following questions in your report:

- Are your results stated clearly and in a form that can be easily presented?
- Are there particularly novel or unique findings that should be highlighted?
- Can you rank the models and findings in order of their applicability to the business goals?
- In general, how well do these results answer your organization's business goals?
- What additional questions have your results raised? How might you phrase these questions in business terms?

After you have evaluated the results, compile a list of approved models for inclusion in the final report. This list should include models that satisfy both the data mining and business goals of your organization.

- [E-Retail Example--Evaluating Results](#)
-

E-Retail Example--Evaluating Results

A Web-Mining Scenario Using CRISP-DM

The overall results of the e-retailer's first experience with data mining are fairly easy to communicate from a business perspective: the study produced what are hoped to be better product recommendations and an improved site design. The improved site design is based on the customer browsing sequences, which show the site features that customers want but require several steps to reach. The evidence that the product recommendations are better is more difficult to convey, because the decision rules can become complicated. To produce the final report, the analysts will try to identify some general trends in the rulesets that can be more easily explained.

Ranking the Models. Because several of the initial models seemed to make business sense, ranking within that group was based on statistical criteria, ease of interpretation, and diversity. Thus, the model gave different recommendations for different situations.

New Questions. The most important question to come out of the study is, How can the e-retailer find out more about his or her customers? The information in the customer database plays an important role in forming the clusters for recommendations. While special rules are available for making recommendations to customers whose information is missing, the recommendations are more general in nature than those that can be made to registered customers.

Review Process

Effective methodologies usually include time for reflection on the successes and weaknesses of the process just completed. Data mining is no different. Part of CRISP-DM is learning from your experience so that future data mining projects will be more effective.

Task List

First, you should summarize the activities and decisions for each phase, including data preparation steps, model building, etc. Then for each phase, consider the following questions and make suggestions for improvement:

- Did this stage contribute to the value of the final results?
 - Are there ways to streamline or improve this particular stage or operation?
 - What were the failures or mistakes of this phase? How can they be avoided next time?
 - Were there dead ends, such as particular models that proved fruitless? Are there ways to predict such dead ends so that efforts can be directed more productively?
 - Were there any surprises (both good and bad) during this phase? In hindsight, is there an obvious way to predict such occurrences?
 - Are there alternative decisions or strategies that might have been used in a given phase? Note such alternatives for future data mining projects.
- [E-Retail Example--Review Report](#)
-

E-Retail Example--Review Report

A Web-Mining Scenario Using CRISP-DM

As a result of reviewing the process of the initial data mining project, the e-retailer has developed a greater appreciation of the interrelations between steps in the process. Initially reluctant to "backtrack" in the CRISP-DM process, the e-retailer now sees that the [cyclic nature of the process](#) increases its power. The process review has also led the e-retailer to understand that:

- A return to the exploration process is always warranted when something unusual appears in another phase of the CRISP-DM process.
 - Data preparation, especially of Web logs, requires patience, since it can take a very long time.
 - It is vital to stay focused on the business problem at hand, because once the data are ready for analysis, it's all too easy to start constructing models without regard to the bigger picture.
 - Once the modeling phase is over, business understanding is even more important in deciding how to implement results and determine what further studies are warranted.
-

Determining the Next Steps

By now, you've produced results, evaluated your data mining experiences, and may be wondering, **Where to next?** This phase helps you to answer that question in light of your business goals for data mining. Essentially, you have two choices at this point:

- **Continue to the deployment phase.** The next phase will help you to incorporate the model results into your business process and produce a final report. Even if your data mining efforts were unsuccessful, you should use the deployment phase of CRISP-DM to create a final report for distribution to the project sponsor.
- **Go back and refine or replace your models.** If you find that your results are almost, but not quite, optimal, consider another round of modeling. You can take what you've learned in this phase and use it to refine the models and produce better results.

Your decision at this point involves the accuracy and relevancy of the modeling results. If the results address your data mining and business goals, then you are ready for the deployment phase. Whatever decision you make, be sure to document the evaluation process thoroughly.

- [E-Retail Example--Next Steps](#)
-

E-Retail Example--Next Steps

A Web-Mining Scenario Using CRISP-DM

The e-retailer is fairly confident of both the accuracy and relevancy of the project results and so is continuing to the deployment phase.

At the same time, the project team is also ready to go back and augment some of the models to include predictive techniques. At this point, they're waiting for delivery of the final reports and a green light from the decision makers.

Deployment

- [Deployment Overview](#)
 - [Planning for Deployment](#)
 - [Planning Monitoring and Maintenance](#)
 - [Producing a Final Report](#)
 - [Conducting a Final Project Review](#)
-

Deployment Overview

Deployment is the process of using your new insights to make improvements within your organization. This can mean a formal integration such as the implementation of a IBM® SPSS® Modeler model producing churn scores that are then read into a data warehouse. Alternatively, deployment can mean that you use the insights gained from data mining to elicit change in your organization. For example, perhaps you discovered alarming patterns in your data indicating a shift in behavior for customers over the age of 30. These results may not be formally integrated into your information systems, but they will undoubtedly be useful for planning and making marketing decisions.

In general, the deployment phase of CRISP-DM includes two types of activities:

- Planning and monitoring the deployment of results
- Completing wrap-up tasks such as producing a final report and conducting a project review

Depending on your organization's requirements, you may need to complete one or both of these steps.

Related information

- [CRISP-DM Help Overview](#)
 - [Planning for Deployment](#)
 - [Planning Monitoring and Maintenance](#)
 - [Producing a Final Report](#)
 - [Conducting a Final Project Review](#)
 - [Business Understanding Overview](#)
 - [Data Understanding Overview](#)
 - [Data Preparation Overview](#)
 - [Modeling Overview](#)
 - [Evaluation Overview](#)
-

Planning for Deployment

Although you may be anxious to share the fruits of your data mining efforts, take time to plan for a smooth and comprehensive deployment of results.

Task List

- The first step is to summarize your results--both models and findings. This helps you determine which models can be integrated within your database systems and which findings should be presented to your colleagues.
- For each deployable model, create a step-by-step plan for deployment and integration with your systems. Note any technical details such as database requirements for model output. For example, perhaps your system requires that modeling output be deployed in a tab-delimited format.
- For each conclusive finding, create a plan to disseminate this information to strategy makers.
- Are there alternative deployment plans for both types of results that are worth mentioning?
- Consider how the deployment will be monitored. For example, how will a model deployed using IBM® SPSS® Modeler Solution Publisher be updated? How will you decide when the model is no longer applicable?
- Identify any deployment problems and plan for contingencies. For example, decision makers may want more information on modeling results and may require that you provide further technical details.
- [E-Retail Example--Deployment Planning](#)

Related information

- [Deployment Overview](#)
- [Planning Monitoring and Maintenance](#)
- [Producing a Final Report](#)
- [Conducting a Final Project Review](#)

E-Retail Example--Deployment Planning

A Web-Mining Scenario Using CRISP-DM

A successful deployment of the e-retailer's data mining results requires that the right information reaches the right people.

Decision makers. Decision makers need to be informed of the recommendations and proposed changes to the site, and provided with short explanations of how these changes will help. Assuming that they accept the results of the study, the people who will implement the changes need to be notified.

Web developers. People who maintain the Web site will have to incorporate the new recommendations and organization of site content. Inform them of what changes *could* happen because of future studies, so they can lay the groundwork now. Getting the team prepared for on-the-fly site construction based upon real-time sequence analysis might be helpful later.

Database experts. The people who maintain the customer, purchase, and product databases should be kept apprised of how the information from the databases is being used and what attributes may be added to the databases in future projects.

Above all, the project team needs to keep in touch with each of these groups to coordinate the deployment of results and planning for future projects.

Planning Monitoring and Maintenance

In a full-fledged deployment and integration of modeling results, your data mining work may be ongoing. For example, if a model is deployed to predict sequences of e-basket purchases, this model will likely need to be evaluated periodically to ensure its effectiveness and to make continuous improvements. Similarly, a model deployed to increase customer retention among high-value customers will likely need to be tweaked once a particular level of retention is reached. The model might then be modified and re-used to retain customers at a lower but still profitable level on the value pyramid.

Task List

Take notes on the following issues and be sure to include them in the final report.

- For each model or finding, which factors or influences (such as market value or seasonal variation) need to be tracked?
- How can the validity and accuracy of each model be measured and monitored?
- How will you determine when a model has "expired"? Give specifics on accuracy thresholds or expected changes in data, etc.
- What will occur when a model expires? Can you simply rebuild the model with newer data or make slight adjustments? Or will changes be pervasive enough as to require a new data mining project?
- Can this model be used for similar business issues once it has expired? This is where good documentation becomes critical for assessing the business purpose for each data mining project.
- [E-Retail Example--Monitoring and Maintenance](#)

Related information

- [Deployment Overview](#)
- [Planning for Deployment](#)
- [Producing a Final Report](#)
- [Conducting a Final Project Review](#)

E-Retail Example--Monitoring and Maintenance

A Web-Mining Scenario Using CRISP-DM

The immediate task for monitoring is to determine whether the new site organization and improved recommendations actually work. That is, are users able to take more direct routes to the pages that they're looking for? Have cross-sales of recommended items increased? After a few weeks of monitoring, the e-retailer will be able to determine the success of the study.

What can be handled automatically is the inclusion of new registered users. When customers register with the site, the current rulesets can be applied to their information to determine what recommendations they should be given.

Deciding when to update the rulesets for determining recommendations is a trickier task. Updating the rulesets is not an automatic process because cluster creation requires human input regarding the appropriateness of a given cluster solution.

As future projects generate more complex models, the need for and amount of monitoring will almost surely increase. When possible, the bulk of the monitoring should be automatic with regularly scheduled reports available for review. Alternatively, the creation of models that provide predictions on the fly may be a direction the company would like to take. This requires more sophistication from the team than the first data mining project.

Producing a Final Report

Writing a final report not only ties up loose ends in earlier documentation, it can also be used to communicate your results. While this may seem straightforward, it's important to present your results to the various people with a stake in the results. This can include both technical administrators who will be responsible for implementation of the modeling results as well as marketing and management sponsors who will make decisions based on your results.

Task List

First, consider the audience of your report. Are they technical developers or market-focused managers? You may need to create separate reports for each audience if their needs are disparate. In either case, your report should include most of the following points:

- A thorough description of the original business problem
- The process used to conduct data mining
- Costs of the project
- Notes on any deviations from the original project plan
- A summary of data mining results, both models and findings
- An overview of the proposed plan for deployment
- Recommendations for further data mining work, including interesting leads discovered during exploration and modeling
- [Preparing a Final Presentation](#)
- [E-Retail Example--Final Report](#)

Related information

- [Deployment Overview](#)
 - [Planning for Deployment](#)
 - [Planning Monitoring and Maintenance](#)
 - [Conducting a Final Project Review](#)
-

Preparing a Final Presentation

In addition to the project report, you may also need to present the project findings to a team of sponsors or related departments. If this is the case, you could use much of the same information in your report but presented from a broader perspective. The charts and graphs in IBM® SPSS® Modeler can easily be exported for this type of presentation.

E-Retail Example--Final Report

A Web-Mining Scenario Using CRISP-DM

The greatest deviation from the original project plan is also an interesting lead for further data mining work. The original plan called for finding out how to have customers spend more time and see more pages on the site per visit.

As it turns out, having a happy customer is not simply a matter of keeping them online. Frequency distributions of time spent per session, split on whether the session resulted in a purchase, found that the session times for most sessions resulting in purchases fall between the session times for two clusters of nonpurchase sessions.

Now that this is known, the issue is to find out whether these customers who spend a long time on the site without purchasing are just browsing or simply can't find what they're looking for. The next step is to find out how to deliver what they're looking for in order to encourage purchases.

Conducting a Final Project Review

This is the final step of the CRISP-DM methodology, and it offers you a chance to formulate your final impressions and collate the lessons learned during the data mining process.

Task List

You should conduct a brief interview with those significantly involved in the data mining process. Questions to consider during these interviews include the following:

- What are your overall impressions of the project?
- What did you learn during the process--both about data mining in general and the data available?
- Which parts of the project went well? Where did difficulties arise? Was there information that might have helped ease the confusion?

After the data mining results have been deployed, you might also interview those affected by the results such as customers or business partners. Your goal here should be to determine whether the project was worthwhile and offered the benefits it set out to create.

The results of these interviews can be summarized along with your own impressions of the project in a final report that should focus on the lessons learned from the experience of mining your stores of data.

- [E-Retail Example--Final Review](#)

Related information

- [Deployment Overview](#)
 - [Planning for Deployment](#)
 - [Planning Monitoring and Maintenance](#)
 - [Producing a Final Report](#)
-

E-Retail Example--Final Review

A Web-Mining Scenario Using CRISP-DM

Project member interviews. The e-retailer finds that project members most closely associated with the study from start to finish are for the most part enthusiastic about the results and look forward to future projects. The database group seems cautiously optimistic; while they appreciate the usefulness of the study, they point out the added burden on database resources. A consultant was available during the study, but going forward, another employee dedicated to database maintenance will be necessary as the scope of the project expands.

Customer interviews. Customer feedback has been largely positive so far. One issue that was not well thought out was the impact of the site design change on established customers. After a few years, the registered customers developed certain expectations about how the site is organized. Feedback from registered users is not quite as positive as from nonregistered customers, and a few greatly dislike the changes. The e-retailer needs to stay aware of this issue and carefully consider whether a change will bring in enough new customers to risk losing existing ones.

AEQMC0000E AEQMC0000E: Failed to create thread.

AEQMC0001E AEQMC0001E: Failed to impersonate user {0}. Error = {1}

AEQMC0002E AEQMC0002E: Failed to make the socket inheritable. Error = {0}

AEQMC0003E AEQMC0003E: Failed to create environment block. Error = {0}

AEQMC0004E AEQMC0004E: Failed to create client: {0} ({1})

AEQMC0005E AEQMC0005E: Bad magic number: {0}

AEQMC0006E AEQMC0006E: Failed to open server socket ({0})

AEQMC0007I AEQMC0007I: Session {0} ({1}) started

AEQMC0008I AEQMC0008I: User {0} authenticated.

AEQMC0009I AEQMC0009I: About to run inline

AEQMC0010I AEQMC0010I: About to run threaded

AEQMC0011I AEQMC0011I: About to run process

AEQMC0012I AEQMC0012I: About to create process as user: {0}

AEQMC0013I AEQMC0013I: About to create process: {0}

AEQMC0014I AEQMC0014I: Process created

AEQMC0015E AEQMC0015E: Process creation failed

AEQMC0016I AEQMC0016I: Environment block destroyed

AEQMC0017E AEQMC0017E: Failed to create session process. Error = {0}

AEQMC0018I AEQMC0018I: Waiting for process to terminate

AEQMC0019I AEQMC0019I: Waiting returned with value of {0}

AEQMC0020I AEQMC0020I: Received exit code {0}

AEQMC0021E AEQMC0021E: Wait timed out

AEQMC0022E AEQMC0022E: Wait failed with error code {0}

AEQMC0023I AEQMC0023I: Process terminated

AEQMC0024I AEQMC0024I: Locking sessions

AEQMC0025I AEQMC0025I: Sessions locked

AEQMC0026I AEQMC0026I: Creating user info

AEQMC0027I AEQMC0027I: Unlocking sessions

AEQMC0028I AEQMC0028I: Sessions unlocked

AEQMC0029I AEQMC0029I: About to create server socket

AEQMC0030I AEQMC0030I: About to initialise server

AEQMC0031E AEQMC0031E: Session {0} terminated with an exception.

AEQMC0032I AEQMC0032I: Session {0} ended

AEQMC0033E AEQMC0033E: Login failed for user: {0}

AEQMC0034I AEQMC0034I: Login succeeded for user: {0}

AEQMC0035I AEQMC0035I: {0}

AEQMC0035E AEQMC0035E: E3008: () ERROR SDLSessionServer

AEQMC0036E AEQMC0036E: Unable to report the following to server from non-server thread: {0}

AEQMC0037E AEQMC0037E: Error occurred while trying to access user defined temp directory: {0}. Temp directory information: {1}

AEQMC0038I AEQMC0038I: Database connection {0} closed

AEQMC0039I AEQMC0039I: Attempting connection to datasource {0}

AEQMC0040I AEQMC0040I: Attempting connection to {0} datasource {1}

AEQMC0041I AEQMC0041I: Database connection {0} ({1}) opened

AEQMC0042I AEQMC0042I: ODBC SQL: {0}

AEQMC0043I AEQMC0043I: Native SQL: {0}

AEQMC0044E AEQMC0044E: Error: {0}

AEQMC0045E AEQMC0045E: Loading SSO component failed on path {0}.

AEQMC0046E AEQMC0046E: Get process address failed for sso_auth_win@ssocomponent library.

AEQMC0047E AEQMC0047E: Get process address failed for sso_auth_unix@ssocomponent library.

AEQMC0048E AEQMC0048E: Get Java VM failed.

AEQMC0049I AEQMC0049I: Forcing termination of session {0}

AEQMC0050I AEQMC0050I: Session starting

AEQMC0051I AEQMC0051I: Root (working) directory set to {0}

AEQMC0052I AEQMC0052I: {0} service started.

AEQMC0053I AEQMC0053I: Hostname: {0}, Operating System: {1}

AEQMC0054I AEQMC0054I: {0} MB physical memory available

AEQMC0055I AEQMC0055I: Max data size = unlimited

AEQMC0056I AEQMC0056I: Max data size = {0} MB

AEQMC0057I AEQMC0057I: Max address space size = unlimited

AEQMC0058I AEQMC0058I: Max address space size = {0} MB

AEQMC0059I AEQMC0059I: {0} MB usable physical memory available

AEQMC0060I AEQMC0060I: {0} processors available

AEQMC0061I AEQMC0061I: Listening on port {0}

AEQMC0062E AEQMC0062E: Failed to accept connection: accept(): error = {0}

AEQMC0063I AEQMC0063I: {0} service stopped.

AEQMC0064E AEQMC0064E: Server init failed. Cannot set directory to: {0}

AEQMC0065E AEQMC0065E: Server init failed: configuration error {0}

AEQMC0066E AEQMC0066E: Server init failed: configuration error {0}

AEQMC0067E AEQMC0067E: Server init failed in method {0}

AEQMC0068E AEQMC0068E: Server init failed in method {0}; error = {1}

AEQMC0069E AEQMC0069E: Failed to set locale to '{0}'

AEQMC0070I AEQMC0070I: Locale set to {0}. Encoding set to {1}

AEQMC0071I AEQMC0071I: Configuration custom overrides enabled.

AEQMC0072I AEQMC0072I: Configuration custom overrides disabled.

AEQMC0073E AEQMC0073E: Failed to make the socket not inheritable

AEQMC0074E AEQMC0074E: Refused connection from {0} (max_sessions exceeded)

AEQMC0075E AEQMC0075E: Accepted connection from {0}

AEQMC0076I AEQMC0076I: Session ended.

AEQMC0077E AEQMC0077E: Failed to create pipe to communicate with COP process. COP disabled.

AEQMC0078E AEQMC0078E: {0}

AEQMC0079I AEQMC0079I: COP client: Updated configuration file: {0}

AEQMC0080E AEQMC0080E: COP client: Failed to update configuration file: {0}

AEQMC0081I AEQMC0081I: {0}

AEQMC0082E AEQMC0082E: Modelerrun init failed: argument string conversion errors

AEQMC0083E AEQMC0083E: Modelerrun init failed: configuration error

AEQMC0084E AEQMC0084E: Failed to open config file: {0}

AEQMC0085E AEQMC0085E: Error reading config file: {0}

AEQMC0086E AEQMC0086E: Failed to backup config file: {0}

AEQMC0087E AEQMC0087E: Failed to update config file: {0}

AEQMC0088E AEQMC0088E: COP Client: Invalid configuration

AEQMC0089E AEQMC0089E: COP Client: Connected to server: {0}

AEQMC0090E AEQMC0090E: COP Client: Failed to connect to server: {0}

AEQMC0091I AEQMC0091I: COP Client: Started

AEQMC0092E AEQMC0092E: COP Client: Failed to Start

AEQMC0093E AEQMC0093E: COP Client: Update failed

AEQMC0094E AEQMC0094E: COP Client: Exception: {0}

AEQMC0095I AEQMC0095I: File {0} closed

AEQMC0096I AEQMC0096I: Attempting to open/create file {0}

AEQMC0097I AEQMC0097I: File {0} ({1}) opened

AEQMC0098I AEQMC0098I: Connected to datasource, retrieved SQL_DBMS_NAME={0}

AEQMC0099E AEQMC0099E: Native SQL Error with Return Code: {0}

AEQMC0132I AEQMC0132I: E3008: () SDLSessionServer.ClientError

AEQMJ0002I AEQMJ0002I: Shutting down application

AEQMJ0003E AEQMJ0003E: The field "{0}" has a role that is unknown or no longer supported. Results from this model should be verified.

AEQMJ0004E AEQMJ0004E: Failed to connect to server

AEQMJ0005E AEQMJ0005E: Failed to copy data into node "{0}"

About IBM SPSS Modeler Text Analytics

IBM® SPSS® Modeler Text Analytics offers powerful text analytic capabilities, which use advanced linguistic technologies and Natural Language Processing (NLP) to rapidly process a large variety of unstructured text data and, from this text, extract and organize the key concepts. Furthermore, IBM SPSS Modeler Text Analytics can group these concepts into categories.

Around 80% of data held within an organization is in the form of text documents—for example, reports, Web pages, e-mails, and call center notes. Text is a key factor in enabling an organization to gain a better understanding of their customers' behavior. A system that incorporates NLP can intelligently extract concepts, including compound phrases. Moreover, knowledge of the underlying language allows classification of terms into related groups, such as products, organizations, or people, using meaning and context. As a result, you can quickly determine the relevance of the information to your needs. These extracted concepts and categories can be combined with existing structured data, such as demographics, and applied to modeling in IBM SPSS Modeler's full suite of data mining tools to yield better and more-focused decisions.

Linguistic systems are knowledge sensitive—the more information contained in their dictionaries, the higher the quality of the results. IBM SPSS Modeler Text Analytics is delivered with a set of linguistic resources, such as dictionaries for terms and synonyms, libraries, and templates. This product further allows you to develop and refine these linguistic resources to your context. Fine-tuning of the linguistic resources is often an iterative process and is necessary for accurate concept retrieval and categorization. Custom templates, libraries, and dictionaries for specific domains, such as CRM and genomics, are also included.

Deployment. You can deploy text mining streams using the IBM SPSS Modeler Solution Publisher for real-time scoring of unstructured data. The ability to deploy these streams ensures successful, closed-loop text mining implementations. For example, your organization can now analyze scratch-pad notes from inbound or outbound callers by applying your predictive models to increase the accuracy of your marketing message in real time.

To run IBM SPSS Modeler Text Analytics with IBM SPSS Modeler Solution Publisher, add the directory `<install_directory>/ext/bin/spss.TMWBServer` to the `$LD_LIBRARY_PATH` environment variable.

Note: The Japanese adapter for IBM SPSS Modeler Text Analytics has been deprecated starting with version 18.1.

- [Upgrading IBM SPSS Modeler Text Analytics](#)
- [About text mining](#)
- [IBM SPSS Modeler Text Analytics nodes](#)
- [Applications](#)

Upgrading IBM SPSS Modeler Text Analytics

Before installing IBM® SPSS® Modeler Text Analytics, you should save and export any TAPs, templates, and libraries from your current version that you want to use in the new version. We recommend that you save these files to a directory that will not get deleted or overwritten when you install the latest version.

After you install the latest version of IBM SPSS Modeler Text Analytics, you can load the saved TAP file, add any saved libraries, or import and load any saved templates to use them in the latest version.

Important: If you uninstall your current version without saving and exporting the files you require first, any TAP, template, and public library work performed in the previous version will be lost and unable to be used in the latest version of IBM SPSS Modeler Text Analytics.

About text mining

Today an increasing amount of information is being held in unstructured and semistructured formats, such as customer e-mails, call center notes, open-ended survey responses, news feeds, Web forms, etc. This abundance of information poses a problem to many organizations that ask themselves, "How can we collect, explore, and leverage this information?"

Text mining is the process of analyzing collections of textual materials in order to capture key concepts and themes and uncover hidden relationships and trends without requiring that you know the precise words or terms that authors have used to express those concepts. Although they are quite different, text mining is sometimes confused with information retrieval. While the accurate retrieval and storage of information is an enormous challenge, the extraction and management of quality content, terminology, and relationships contained within the information are crucial and critical processes.

Text mining and data mining

For each article of text, linguistic-based text mining returns an index of concepts, as well as information about those concepts. This distilled, structured information can be combined with other data sources to address questions such as:

- Which concepts occur together?
- What else are they linked to?
- What higher level categories can be made from extracted information?
- What do the concepts or categories predict?
- How do the concepts or categories predict behavior?

Combining text mining with data mining offers greater insight than is available from either structured or unstructured data alone. This process typically includes the following steps:

1. **Identify the text to be mined.** Prepare the text for mining. If the text exists in multiple files, save the files to a single location. For databases, determine the field containing the text.
2. **Mine the text and extract structured data.** Apply the text mining algorithms to the source text.
3. **Build concept and category models.** Identify the key concepts and/or create categories. The number of concepts returned from the unstructured data is typically very large. Identify the best concepts and categories for scoring.
4. **Analyze the structured data.** Employ traditional data mining techniques, such as clustering, classification, and predictive modeling, to discover relationships between the concepts. Merge the extracted concepts with other structured data to predict future behavior based on the concepts.

Text analysis and categorization

Text analysis, a form of qualitative analysis, is the extraction of useful information from text so that the key ideas or concepts contained within this text can be grouped into an appropriate number of categories. Text analysis can be performed on all types and lengths of text, although the approach to the analysis will vary somewhat.

Shorter records or documents are most easily categorized, since they are not as complex and usually contain fewer ambiguous words and responses. For example, with short, open-ended survey questions, if we ask people to name their three favorite vacation activities, we might expect to see many short answers, such as *going to the beach*, *visiting national parks*, or *doing nothing*. Longer, open-ended responses, on the other hand, can be quite complex and very lengthy, especially if respondents are educated, motivated, and have enough time to complete a questionnaire. If we ask people to tell us about their political beliefs in a survey or have a blog feed about politics, we might expect some lengthy comments about all sorts of issues and positions.

The ability to extract key concepts and create insightful categories from these longer text sources in a very short period of time is a key advantage of using IBM® SPSS® Modeler Text Analytics. This advantage is obtained through the combination of automated linguistic and statistical techniques to yield the most reliable results for each stage of the text analysis process.

Linguistic processing and NLP

The primary problem with the management of all of this unstructured text data is that there are no standard rules for writing text so that a computer can understand it. The language, and consequently the meaning, varies for every document and every piece of text. The only way to accurately retrieve and organize such unstructured data is to analyze the language and thus uncover its meaning. There are several different automated approaches to the extraction of concepts from unstructured information. These approaches can be broken down into two kinds, linguistic and nonlinguistic.

Some organizations have tried to employ automated nonlinguistic solutions based on statistics and neural networks. Using computer technology, these solutions can scan and categorize key concepts more quickly than human readers can. Unfortunately, the accuracy of such solutions is fairly low. Most statistics-based systems simply count the number of times words occur and calculate their statistical proximity to related concepts. They produce many irrelevant results, or noise, and miss results they should have found, referred to as silence.

To compensate for their limited accuracy, some solutions incorporate complex nonlinguistic rules that help to distinguish between relevant and irrelevant results. This is referred to as *rule-based text mining*.

Linguistics-based text mining, on the other hand, applies the principles of natural language processing (NLP)—the computer-assisted analysis of human languages—to the analysis of words, phrases, and syntax, or structure, of text. A system that incorporates NLP can intelligently extract concepts, including compound phrases. Moreover, knowledge of the underlying language allows classification of concepts into related groups, such as products, organizations, or people, using meaning and context.

Linguistics-based text mining finds meaning in text much as people do—by recognizing a variety of word forms as having similar meanings and by analyzing sentence structure to provide a framework for understanding the text. This approach offers the speed and cost-effectiveness of statistics-based systems, but it offers a far higher degree of accuracy while requiring far less human intervention.

To illustrate the difference between statistics-based and linguistics-based approaches during the extraction process, consider how each would respond to a query about **reproduction of documents**. Both statistics-based and linguistics-based solutions would have to expand the word **reproduction** to include synonyms, such as **copy** and **duplication**. Otherwise, relevant information will be overlooked. But if a statistics-based solution attempts to do this type of synonymy—searching for other terms with the same meaning—it is likely to include the term **birth** as well, generating a number of irrelevant results. The understanding of language cuts through the ambiguity of text, making linguistics-based text mining, by definition, the more reliable approach.

Understanding how the extraction process works can help you make key decisions when fine-tuning your linguistic resources (libraries, types, synonyms, and more). Steps in the extraction process include:

- Converting source data to a standard format
- Identifying candidate terms
- Identifying equivalence classes and integration of synonyms
- Assigning a type
- Indexing and, when requested, pattern matching with a secondary analyzer

Step 1. Converting source data to a standard format

In this first step, the data you import is converted to a uniform format that can be used for further analysis. This conversion is performed internally and does not change your original data.

Step 2. Identifying candidate terms

It is important to understand the role of linguistic resources in the identification of candidate terms during linguistic extraction. Linguistic resources are used every time an extraction is run. They exist in the form of templates, libraries, and compiled resources. Libraries include lists of words, relationships, and other information used to specify or tune the extraction. The compiled resources cannot be viewed or edited. However, the remaining resources can be edited in the Template Editor or, if you are in an interactive workbench session, in the Resource Editor.

Compiled resources are core, internal components of the extraction engine within IBM SPSS Modeler Text Analytics . These resources include a general dictionary containing a list of base forms with a part-of-speech code (noun, verb, adjective, and so on).

In addition to those compiled resources, several libraries are delivered with the product and can be used to complement the types and concept definitions in the compiled resources, as well as to offer synonyms. These libraries—and any custom ones you create—are made up of several dictionaries. These include type dictionaries, synonym dictionaries, and exclude dictionaries.

Once the data have been imported and converted, the extraction engine will begin identifying candidate terms for extraction. Candidate terms are words or groups of words that are used to identify concepts in the text. During the processing of the text, single words (*uniterms*) and compound words (*multiterms*) are identified using part-of-speech pattern extractors. Then, candidate sentiment keywords are identified using sentiment text link analysis.

Note: The terms in the aforementioned compiled general dictionary represent a list of all of the words that are likely to be uninteresting or linguistically ambiguous as uniterms. These words are excluded from extraction when you are identifying the uniterms. However, they are reevaluated when you are determining parts of speech or looking at longer candidate compound words (multiterms).

Step 3. Identifying equivalence classes and integration of synonyms

After candidate uniterms and multiterms are identified, the software uses a normalization dictionary to identify equivalence classes. An equivalence class is a base form of a phrase or a single form of two variants of the same phrase. The purpose of assigning phrases to equivalence

classes is to ensure that, for example, `side effect` and 副作用 are not treated as separate concepts. To determine which concept to use for the equivalence class—that is, whether `side effect` or 副作用 is used as the lead term—the extraction engine applies the following rules in the order listed:

- The user-specified form in a library.
- The most frequent form, as defined by precompiled resources.

Step 4. Assigning type

Next, types are assigned to extracted concepts. A type is a semantic grouping of concepts. Both compiled resources and the libraries are used in this step. Types include such things as higher-level concepts, positive and negative words, first names, places, organizations, and more. See the topic [Type dictionaries](#) for more information.

Linguistic systems are knowledge sensitive—the more information contained in their dictionaries, the higher the quality of the results. Modification of the dictionary content, such as synonym definitions, can simplify the resulting information. This is often an iterative process and is necessary for accurate concept retrieval. NLP is a core element of IBM SPSS Modeler Text Analytics.

- [How extraction works](#)
- [How categorization works](#)

How extraction works

During the extraction of key concepts and ideas from your responses, IBM® SPSS® Modeler Text Analytics relies on linguistics-based text analysis. This approach offers the speed and cost effectiveness of statistics-based systems. But it offers a far higher degree of accuracy, while requiring far less human intervention. Linguistics-based text analysis is based on the field of study known as natural language processing, also known as computational linguistics.

Understanding how the extraction process works can help you make key decisions when fine-tuning your linguistic resources (libraries, types, synonyms, and more). Steps in the extraction process include:

- Converting source data to a standard format
- Identifying candidate terms
- Identifying equivalence classes and integration of synonyms
- Assigning a type
- Indexing
- Matching patterns and events extraction

Step 1. Converting source data to a standard format

In this first step, the data you import is converted to a uniform format that can be used for further analysis. This conversion is performed internally and does not change your original data.

Step 2. Identifying candidate terms

It is important to understand the role of linguistic resources in the identification of candidate terms during linguistic extraction. Linguistic resources are used every time an extraction is run. They exist in the form of templates, libraries, and compiled resources. Libraries include lists of words, relationships, and other information used to specify or tune the extraction. The compiled resources cannot be viewed or edited. However, the remaining resources (templates) can be edited in the Template Editor or, if you are in an interactive workbench session, in the Resource Editor.

Compiled resources are core, internal components of the extraction engine within IBM SPSS Modeler Text Analytics. These resources include a general dictionary containing a list of base forms with a part-of-speech code (noun, verb, adjective, adverb, participle, coordinator, determiner, or preposition). The resources also include reserved, built-in types used to assign many extracted terms to the following types, `<Location>`, `<Organization>`, or `<Person>`. See the topic [Built-in types](#) for more information.

In addition to those compiled resources, several libraries are delivered with the product and can be used to complement the types and concept definitions in the compiled resources, as well as to offer other types and synonyms. These libraries—and any custom ones you create—are made up of several dictionaries. These include type dictionaries, substitution dictionaries (synonyms and optional elements), and exclude dictionaries. See the topic [Working with Libraries](#) for more information.

Once the data have been imported and converted, the extraction engine will begin identifying candidate terms for extraction. Candidate terms are words or groups of words that are used to identify concepts in the text. During the processing of the text, single words (*uniterms*) that are not in the compiled resources are considered as candidate term extractions. Candidate compound words (*multiterms*) are identified using part-of-speech pattern extractors. For example, the multiterm `sports car`, which follows the "adjective noun" part-of-speech pattern, has two components. The multiterm `fast sports car`, which follows the "adjective adjective noun" part-of-speech pattern, has three components.

Note: The terms in the aforementioned compiled general dictionary represent a list of all of the words that are likely to be uninteresting or linguistically ambiguous as uniterms. These words are excluded from extraction when you are identifying the uniterms. However, they are reevaluated when you are determining parts of speech or looking at longer candidate compound words (multiterms). Finally, a special algorithm is used to handle uppercase letter strings, such as job titles, so that these special patterns can be extracted.

Step 3. Identifying equivalence classes and integration of synonyms

After candidate uniterms and multiterms are identified, the software uses a set of algorithms to compare them and identify equivalence classes. An equivalence class is a base form of a phrase or a single form of two variants of the same phrase. The purpose of assigning phrases to equivalence classes is to ensure that, for example, **president of the company** and **company president** are not treated as separate concepts. To determine which concept to use for the equivalence class—that is, whether **president of the company** or **company president** is used as the lead term, the extraction engine applies the following rules in the order listed:

- The user-specified form in a library.
- The most frequent form in the full body of text.
- The shortest form in the full body of text (which usually corresponds to the base form).

Step 4. Assigning type

Next, types are assigned to extracted concepts. A type is a semantic grouping of concepts. Both compiled resources and the libraries are used in this step. Types include such things as higher-level concepts, positive and negative words, first names, places, organizations, and more. Additional types can be defined by the user. See the topic [Type dictionaries](#) for more information.

Step 5. Indexing

The entire set of records or documents is indexed by establishing a pointer between a text position and the representative term for each equivalence class. This assumes that all of the inflected form instances of a candidate concept are indexed as a candidate base form. The global frequency is calculated for each base form.

Step 6. Matching patterns and events extraction

IBM SPSS Modeler Text Analytics can discover not only types and concepts but also relationships among them. Several algorithms and libraries are available with this product and provide the ability to extract relationship patterns between types and concepts. They are particularly useful when attempting to discover specific opinions (for example, product reactions) or the relational links between people or objects (for example, links between political groups or genomes).

How categorization works

When creating category models in IBM® SPSS® Modeler Text Analytics, there are several different techniques you can choose to create categories. Because every dataset is unique, the number of techniques and the order in which you apply them may change. Since your interpretation of the results may be different from someone else's, you may need to experiment with the different techniques to see which one produces the best results for your text data. In IBM SPSS Modeler Text Analytics, you can create category models in a workbench session in which you can explore and fine-tune your categories further.

In this guide, category building refers to the generation of category definitions and classification through the use of one or more built-in techniques, and categorization refers to the scoring, or labeling, process whereby unique identifiers (name/ID/value) are assigned to the category definitions for each record or document.

During category building, the concepts and types that were extracted are used as the building blocks for your categories. When you build categories, the records or documents are automatically assigned to categories if they contain text that matches an element of a category's definition.

IBM SPSS Modeler Text Analytics offers you several automated category building techniques to help you categorize your documents or records quickly.

Grouping Techniques

Each of the techniques available is well suited to certain types of data and situations, but often it is helpful to combine techniques in the same analysis to capture the full range of documents or records. You may see a concept in multiple categories or find redundant categories.

Concept Root Derivation. This technique creates categories by taking a concept and finding other concepts that are related to it by analyzing whether any of the concept components are morphologically related, or share roots. This technique is very useful for identifying synonymous compound word concepts, since the concepts in each category generated are synonyms or closely related in meaning. It works with data of varying lengths and generates a smaller number of compact categories. For example, the concept **opportunities to**

advance would be grouped with the concepts **opportunity for advancement** and **advancement opportunity**. See the topic [Concept root derivation](#) for more information.

Semantic Network. This technique begins by identifying the possible senses of each concept from its extensive index of word relationships and then creates categories by grouping related concepts. This technique is best when the concepts are known to the semantic network and are not too ambiguous. It is less helpful when text contains specialized terminology or jargon unknown to the network. In one example, the concept **granny smith apple** could be grouped with **gala apple** and **winesap apple** since they are siblings of the **granny smith**. In another example, the concept **animal** might be grouped with **cat** and **kangaroo** since they are hyponyms of **animal**. This technique is available for English text only in this release. See the topic [Semantic Networks](#) for more information.

Concept Inclusion. This technique builds categories by grouping multiterm concepts (compound words) based on whether they contain words that are subsets or supersets of a word in the other. For example, the concept **seat** would be grouped with **safety seat**, **seat belt**, and **seat belt buckle**. See the topic [Concept Inclusion](#) for more information.

Co-occurrence. This technique creates categories from co-occurrences found in the text. The idea is that when concepts or concept patterns are often found together in documents and records, that co-occurrence reflects an underlying relationship that is probably of value in your category definitions. When words co-occur significantly, a co-occurrence rule is created and can be used as a category descriptor for a new subcategory. For example, if many records contain the words **price** and **availability** (but few records contain one without the other), then these concepts could be grouped into a co-occurrence rule, (**price & available**) and assigned to a subcategory of the category **price** for instance. See the topic [Co-occurrence Rules](#) for more information.

Minimum number of documents. To help determine how interesting co-occurrences are, define the minimum number of documents or records that must contain a given co-occurrence for it to be used as a descriptor in a category.

IBM SPSS Modeler Text Analytics nodes

Along with the many standard nodes delivered with IBM® SPSS® Modeler, you can also work with text mining nodes to incorporate the power of text analysis into your streams. IBM SPSS Modeler Text Analytics offers you several text mining nodes to do just that. These nodes are stored in the IBM SPSS Modeler Text Analytics tab of the node palette.

The following nodes are included:

- The File List source node generates a list of document names as input to the text mining process. This is useful when the text resides in external documents rather than in a database or other structured file. The node outputs a single field with one record for each document or folder listed, which can be selected as input in a subsequent Text Mining node. See the topic [File List node](#) for more information.
- The Web Feed source node makes it possible to read in text from Web feeds, such as blogs or news feeds in RSS or HTML formats, and use this data in the text mining process. The node outputs one or more fields for each record found in the feeds, which can be selected as input in a subsequent Text Mining node. See the topic [Web Feed node](#) for more information.
- The Language Identifier node is a process node that scans source text to determine which human language it is written in and then marks that up in a new field. Primarily designed to be used with large amounts of data, this node is particularly useful when you have more than one language in your data sources and want to process just one language. See the topic [Language Node](#) for more information.
- The Text Mining node uses linguistic methods to extract key concepts from the text, allows you to create categories with these concepts and other data, and offers the ability to identify relationships and associations between concepts based on known patterns (called text link analysis). The node can be used to explore the text data contents or to produce either a concept model or category model. The concepts and categories can be combined with existing structured data, such as demographics, and applied to modeling. See the topic [Text Mining modeling node](#) for more information.
- The Text Link Analysis node extracts concepts and also identifies relationships between concepts based on known patterns within the text. Pattern extraction can be used to discover relationships between your concepts, as well as any opinions or qualifiers attached to these concepts. The Text Link Analysis node offers a more direct way to identify and extract patterns from your text and then add the pattern results to the dataset in the stream. But you can also perform TLA using an interactive workbench session in the Text Mining modeling node. See the topic [Text Link Analysis node](#) for more information.
- When mining text from external documents, the Text Mining Output node can be used to generate an HTML page that contains links to the documents from which concepts were extracted. See the topic [File Viewer node](#) for more information.

Applications

In general, anyone who routinely needs to review large volumes of documents to identify key elements for further exploration can benefit from IBM® SPSS® Modeler Text Analytics.

Some specific applications include:

- Scientific and medical research. Explore secondary research materials, such as patent reports, journal articles, and protocol publications. Identify associations that were previously unknown (such as a doctor associated with a particular product), presenting avenues for further exploration. Minimize the time spent in the drug discovery process. Use as an aid in genomics research.

- Investment research. Review daily analyst reports, news articles, and company press releases to identify key strategy points or market shifts. Trend analysis of such information reveals emerging issues or opportunities for a firm or industry over a period of time.
 - Fraud detection. Use in banking and health-care fraud to detect anomalies and discover red flags in large amounts of text.
 - Market research. Use in market research endeavors to identify key topics in open-ended survey responses.
 - Blog and Web feed analysis. Explore and build models using the key ideas found in news feeds, blogs, etc.
 - CRM. Build models using data from all customer touch points, such as e-mail, transactions, and surveys.
-

Reading in Source Text

Data for text mining can be in any of the standard formats that are used by IBM® SPSS® Modeler, including databases or other "rectangular" formats that represent data in rows and columns, or in document formats, such as Microsoft Word, Adobe PDF, or HTML, that do not conform to this structure.

- To read in text from documents that do not conform to standard data structure, including Microsoft Word, Microsoft Excel, and Microsoft PowerPoint, in addition to Adobe PDF, XML, HTML, and others, the File List node can be used to generate a list of documents or folders as input to the text mining process. For more information, see [File List node](#).
 - To read in text from web feeds, such as blogs or news feeds in RSS or HTML formats, the Web Feed node can be used to format web feed data for input into the text mining process. For more information, see [Web Feed node](#).
 - To read in text from any of the standard data formats used by SPSS Modeler, such as a database with one or more text fields for customer comments, you can use any of the SPSS Modeler source nodes. For more information, see the SPSS Modeler node documentation.
 - When you are processing large amounts of data, which might include text in several different languages, use the Language node to identify the language used in a specific field. For more information, see [Language Node](#).
- [File List node](#)
 - [Web Feed node](#)
 - [Language Node](#)

Related information

- [File List node](#)
 - [Web Feed node](#)
 - [Language Node](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

File List node

To read in text from unstructured documents saved in formats such as Microsoft Word, Microsoft Excel, and Microsoft PowerPoint, as well as Adobe PDF, XML, HTML, and others, the File List node can be used to generate a list of documents or folders as input to the text mining process. This is necessary because unstructured text documents cannot be represented by fields and records—rows and columns—in the same manner as other data used by IBM® SPSS® Modeler.

The File List node functions as a source node.

You can find this node on the IBM SPSS Modeler Text Analytics tab of nodes palette at the bottom of the IBM SPSS Modeler window. See the topic [IBM SPSS Modeler Text Analytics nodes](#) for more information.

Important: Any directory names and file names containing characters that are not included in the machine local encoding are not supported. When attempting to execute a stream containing a File List node, any file or directory names containing these characters will cause the stream execution to fail. This can happen with foreign language directory names or file names, such as a German filename on a French locale.

Local data support. If you are connected to a remote IBM SPSS Modeler Text Analytics Server and have a stream with a File List node, the data should reside on the same machine as the IBM SPSS Modeler Text Analytics Server – or ensure that the server machine has access to the folder where the source data in the File List node is stored.

Note: You cannot use the File List node for scoring within an IBM SPSS Collaboration and Deployment Services - Scoring configuration.

- [File List node: Settings tab](#)
 - [File List Node: Other Tabs](#)
 - [Using the File List node in text mining](#)
-

File List node: Settings tab

On this tab you define the directories, file extensions, and input for this node.

Note: Text mining extraction cannot process Microsoft Office and Adobe PDF files under non-Microsoft Windows platforms. However, XML, HTML or text files can always be processed.

Any directory names and file names containing characters that are not included in the machine local encoding are not supported. When attempting to execute a stream containing a File List node, any file or directory names containing these characters will cause the stream execution to fail. This can happen with foreign language directory names or file names, such as a German filename on a French locale.

Directory. Specifies the root folder containing the documents that you want to list.

- Include subdirectories. Specifies that subdirectories should also be scanned.

File type(s) to include in list: You can select or deselect the file types and extensions you want to use. By deselecting a file extension, the files with that extension are ignored. You can filter by the following extensions:

Table 1. File type filters by file extension

• .rtf, .doc, .docx, .docm	• .xls, .xlsx, .xlsm	• .ppt, .pptx, .pptm	• .txt, .text
• .htm, .html, .shtml	• .xml	• .pdf	• .\$

Note: For more information, see [File List node](#).

If you have files with either no extension, or a trailing dot extension (for example `File01` or `File01.`), use the No extension option to select them.

Only outputs document pathnames. Select this option if the output field will contain one or more pathnames for the location(s) of where the documents reside.

Input encoding. If the output field will contain exact text, choose the relevant value from the following list:

- Automatic (European)
- UTF-8
- UTF-16
- ISO-8859-1
- ISO-8859-2
- Windows-1250
- US ascii

The output is shown as UTF-8 document text.

File List Node: Other Tabs

The Types tab is a standard tab in IBM® SPSS® Modeler nodes, as is the Annotations tab.

Related information

- [File List node](#)
- [File List node: Settings tab](#)
- [Using the File List node in text mining](#)
- [File List Node: filelistnode](#)
- [Microsoft Internet Explorer settings for Help](#)

Using the File List node in text mining

The File List node is used when the text data resides in external unstructured documents in formats such as Microsoft Word, Microsoft Excel, and Microsoft PowerPoint, as well as Adobe PDF, XML, HTML, and others.

As an example, suppose we connected a File List node to a Text Mining node in order to supply text that resides in external documents:

1. **File List node (Settings tab).** First, we added this node to the stream to specify where the text documents are stored. We selected the directory containing all of the documents on which we want to perform text mining.
2. **Text Mining node (Fields tab).** Next, we added and connected a Text Mining node to the File List node. In this node, we defined our input format, resource template, and output format. We selected the field name produced from the File List node, the text field, and other settings. See the topic [Using the Text Mining node in a stream](#) for more information.

For more information on using the Text Mining node, see [Text Mining modeling node](#).

Web Feed node

The Web Feed node can be used to prepare text data from Web feeds for the text mining process. This node accepts Web feeds in two formats:

- RSS Format. RSS is a simple XML-based standardized format for Web content. The URL for this format points to a page that has a set of linked articles such as syndicated news sources and blogs. Since RSS is a standardized format, each linked article is automatically identified and treated as a separate record in the resulting data stream. No further input is required for you to be able to identify the important text data and the records from the feed unless you want to apply a filtering technique to the text.
- HTML Format. You can define one or more URLs to HTML pages on the Input tab. Then, in the Records tab, define the record start tag as well as identify the tags that delimit the target content and assign those tags to the output fields of your choice (description, title, modified date, and so on). See the topic [Web Feed Node: Records Tab](#) for more information.

Important! If you are trying to retrieve information over the web through a proxy server, you must enable the proxy server in the `net.properties` file for both the IBM® SPSS® Modeler Text Analytics Client and Server. Follow the instructions detailed inside this file. This applies when accessing the web through the Web Feed node or retrieving an SDL Software as a Service (SaaS) license since these connections go through Java™. This file is located in `C:\Program Files\IBM\SPSS\Modeler\18.4.0\jre\lib\net.properties` by default.

The output of this node is a set of fields used to describe the records. The Description field is most commonly used since it contains the bulk of the text content. However, you may also be interested in other fields, such as the short description of a record (Short Desc field) or the record's title (Title field). Any of the output fields can be selected as input for a subsequent Text Mining node.

Note: You cannot use the Web Feed node for scoring within an IBM SPSS Collaboration and Deployment Services - Scoring configuration. You can find this node on the IBM SPSS Modeler Text Analytics tab of nodes palette at the bottom of the IBM SPSS Modeler window. See the topic [IBM SPSS Modeler Text Analytics nodes](#) for more information.

- [Web Feed Node: Input Tab](#)
- [Web Feed Node: Records Tab](#)
- [Web Feed Node: Content Filter Tab](#)
- [Using the Web Feed Node in Text Mining](#)

Web Feed Node: Input Tab

The Input tab is used to specify one or more Web addresses, or URLs, in order to capture the text data. In the context of text mining, you could specify URLs for feeds that contain text data.

Important: When working with non RSS data, you may prefer to use a web scraping tool, such as WebQL®, to automate content gathering and then referring the output from that tool using a different source node.

You can set the following parameters:

Enter or paste URLs. In this field, you can type or paste one or more URLs. If you are entering more than one, enter only one per line and use the Enter/Return key to separate lines. Enter the full URL path to the file. These URLs can be for feeds in one of two formats:

- RSS format. RSS is a simple XML-based standardized format for Web content. The URL for this format points to a page that has a set of linked articles such as syndicated news sources and blogs. Since RSS is a standardized format, each linked article is automatically identified and treated as a separate record in the resulting data stream. No further input is required for you to be able to identify the important text data and the records from the feed unless you want to apply a filtering technique to the text.
- HTML format. You can define one or more URLs to HTML pages on the Input tab. Then, in the Records tab, define the record start tag as well as identify the tags that delimit the target content and assign those tags to the output fields of your choice (description, title, modified date, and so on). When working with non RSS data, you may prefer to use a web scraping tool, such as WebQL®, to automate content gathering and then referring the output from that tool using a different source node. See the topic [Web Feed Node: Records Tab](#) for more information.

Number of most recent entries to read per URL. This field specifies the maximum number of records to read for each URL listed in the field starting with the first record found in the feed. The amount of text impacts the processing speed during extraction downstream in a Text Mining node or Text Link Analysis node.

Save and reuse previous web feeds when possible. With this option, web feeds are scanned and the processed results are cached. Then, upon subsequent stream executions, if the contents of a given feed have not changed or if the feed is inaccessible (an Internet outage, for example), the cached version is used to speed processing time. Any new content discovered in these feeds is also cached for the next time you execute the node.

- Label. If you select Save and reuse previous web feeds when possible, you must specify a label name for the results. This label is used to describe the cached feeds on the server. If no label is specified or the label is unrecognized, no reuse will be possible.

Web Feed Node: Records Tab

The Records tab is used to specify the text content of non-RSS feeds by identifying where each new record begins, as well as other relevant information regarding each record. If you know that a non-RSS feed (HTML) contains text that is in multiple records, you must identify the record start tag here or else the text will be treated as one record. While RSS feeds are standardized and do not require any tag specification on this tab, you can still preview the content in the Preview tab.

Important: When working with non RSS data, you may prefer to use a web scraping tool, such as WebQL®, to automate content gathering and then referring the output from that tool using a different source node.

URL. This drop-down list contains a list of URLs entered on the Input tab. Both HTML and RSS formatted feeds are present. If the URL address is too long for the drop-down list, it will automatically be clipped in the middle using an ellipsis to replace the clipped text, such as <http://www.ibm.com/example/start-of-address...rest-of-address/path.htm>.

- With HTML formatted feeds, if the feed contains more than one record (or entry), you can define which HTML tags contain the data corresponding to the field shown in the table. For example, you can define the start tag that indicates a new record has started, a modified date tag, or an author name.
- With RSS formatted feeds, you are not prompted to enter any tags since RSS is a standardized format. However, you can view sample results on the Preview tab if desired. All recognized RSS feeds are preceded by the RSS logo image.

Source tab. On this tab, you can view the source code for any HTML feeds. This code is not editable. You can use the Find field to locate specific tags or information on this page that you can then copy and paste into the table below. The Find field is not case sensitive and will match partial strings.

Preview tab. On this tab, you can preview how a record will be read by the Web feed node. This is particularly useful for HTML feeds since you can change how a record will be read by defining HTML tags in the table below the Preview tab.

Non-RSS record start tag. This option only applies to non-RSS feeds. If your HTML feed contains multiple text that you want to break up into multiple records, specify the HTML tag that signals the beginning of a record (such as an article or blog entry) here. If you don't define one for a non-RSS feed, Modeler will try to guess the XML format and return corresponding records. If Modeler can't guess the XML format, nothing will be returned. If your goal is to import the whole content of a page and then process it later, we recommend using separate XML readers with more powerful functionality and then import the result into Modeler Text Analytics.

Field table. This option only applies to non-RSS feeds. In this table, you can break up the text content into specific output fields by entering a start tag for any of the predefined output fields. Enter the start tag only. All matches are done by parsing the HTML and matching the table contents to the tag names and attributes found in the HTML. You can use the buttons at the bottom to copy the tags you have defined and reuse them for other feeds.

Table 1. Possible output fields for non-RSS feeds (HTML formats)

Output Field Name	Expected Tag Content
Title	The tag delimiting the record title. (optional)
Short Desc	The tag delimiting the short description or label. (optional)
Description	The tag delimiting the main text. If left blank, this field will contain all other content in either the <body> tag (if there is a single record) or the content found inside the current record (when a record delimiter has been specified).
Author	The tag delimiting the author of the text. (optional)
Contributors	The tag delimiting the names of the contributors. (optional)
Published Date	The tag delimiting the date when the text was published. If left blank, this field will contain the date when the node reads the data.
Modified Date	The tag delimiting the date when the text was modified. If left blank, this field will contain the date when the node reads the data.

When you enter a tag into the table, the feed is scanned using this tag as the minimum tag to match rather than an exact match. That is, if you entered **<div>** for the Title field, this would match any **<div>** tag in the feed, including those with specified attributes (such as **<div class="post three">**), such that **<div>** is equal to the root tag (**<div>**) and any derivative that includes an attribute and use that content for the Title output field. If you enter a root tag, any further attributes are also included.

Table 2. Examples of HTML tags used identify the text for the output fields

If you enter:	It would match:	And also match:	But not match:
<div>	<div>	<div class="post">	any other tag
<p class="auth">	<p class="auth">	<p color="black" class="auth" id="85643">	<p color="black">

Web Feed Node: Content Filter Tab

The Content Filter tab is used to apply a filter technique to RSS feed content. This tab does not apply to HTML feeds. You may want to filter if the feed contains a lot of text in the form of headers, footers, menus, advertising and so on. You can use this tab to strip out unwanted HTML tags, JavaScript, and short words or lines from the content.

Content Filtering. If you do not want to apply a cleaning technique, select None. Otherwise, select RSS Content Cleaner.

RSS Content Cleaner Options. If you select RSS Content Cleaner, you can choose to discard lines based on certain criteria. A line is delimited by an HTML tag such as <p> and but excluding in-line tags such as , , and . Please note that
 tags are processed as line breaks.

- **Discard short lines.** This option ignores lines that do not contain the minimum number of words defined here.
- **Discard lines with short words.** This option ignores lines that have more than the minimum average word length defined here.
- **Discard lines with many single character words.** This option ignores lines that contain more than a certain proportion of single character words.
- **Discard lines containing specific tags.** This option ignores text in lines that contain any of the tags specified in the field.
- **Discard lines containing specific text.** This option ignores lines that contain any of the text specified in the field.

Related information

- [Web Feed node](#)
- [Web Feed Node: Input Tab](#)
- [Web Feed Node: Records Tab](#)
- [Using the Web Feed Node in Text Mining](#)
- [Web Feed Node: webfeednode](#)
- [Microsoft Internet Explorer settings for Help](#)

Using the Web Feed Node in Text Mining

The Web Feed node can be used to prepare text data from Internet Web feeds for the text mining process. This node accepts Web feeds in either an HTML or RSS format. These feeds serve as input into the text mining process (a subsequent Text Mining or Text Link Analysis node).

If you use the Web Feed node, you must make sure to specify that the Text field represents actual text in the Text Mining or Text Link Analysis node to indicate that these feeds link directly to each article or blog entry.

Important! If you are trying to retrieve information over the web through a proxy server, you must enable the proxy server in the `net.properties` file for both the IBM® SPSS® Modeler Text Analytics Client and Server. Follow the instructions detailed inside this file. This applies when accessing the web through the Web Feed node or retrieving an SDL Software as a Service (SaaS) license since these connections go through Java™. This file is located in `C:\Program Files\IBM\SPSS\Modeler\18.4.0\jre\lib\net.properties` by default.

Example: Web Feed node (RSS Feed) with the Text Mining modeling node

As an example, suppose we connect a Web Feed node to a Text Mining node in order to supply text data from an RSS feed into the text mining process.

1. **Web Feed node (Input tab).** First, we added this node to the stream to specify where the feed contents are located and to verify the content structure. On the first tab, we provided the URL to an RSS feed. Since our example is for an RSS feed, the formatting is already defined, and we do not need to make any changes on the Records tab. An optional content filtering algorithm is available for RSS feeds, however in this case it was not applied.
2. **Text Mining node (Fields tab).** Next, we added and connected a Text Mining node to the Web Feed node. On this tab, we defined the text field output by the Web Feed node. In this case, we wanted to use the Description field. We also selected the option Text field represents actual text, as well as other settings.
3. **Text Mining node (Model tab).** Next, on the Model tab, we chose the build mode and resources. In this example, we chose to build a concept model directly from this node using the default resource template.

For more information on using the Text Mining node, see [Text Mining modeling node](#).

Related information

- [Web Feed node](#)
- [Web Feed Node: Input Tab](#)
- [Web Feed Node: Records Tab](#)
- [Web Feed Node: Content Filter Tab](#)
- [Web Feed Node: webfeednode](#)
- [Microsoft Internet Explorer settings for Help](#)

Language Node

You can use the Language node to identify the natural language of a text field within your source data.

The output of this node is a derived field that contains the detected language code.

Note: You cannot use the Language node for scoring within an IBM® SPSS® Collaboration and Deployment Services - Scoring configuration. You can find this node on the IBM SPSS Modeler Text Analytics tab of nodes palette at the bottom of the IBM SPSS Modeler window. See the topic [IBM SPSS Modeler Text Analytics nodes](#) for more information.

- [Language Node: Settings Tab](#)

Language Node: Settings Tab

On this tab you specify how to output the language details for a selected text field.

Text field Select the text field for which you want to identify the language.

Derive field name Enter a name for the derived field which will contain the detected language code. The default value is *Language*.

Default value for when language cannot be identified Specify the name of the field to be created if the language cannot be identified. The available choices are:

- Undefined If selected, the derived field contains null values.
- Supported If selected, you can choose from one of the following supported ISO languages:
 - English (EN)
 - German (DE)
 - Spanish (ES)
 - French (FR)
 - Italian (IT)
 - Dutch (NL)
 - Portuguese (PT)
- Custom If no supported language is suitable, use this option to specify that a custom value should be used. Typically this might be a 2 letter ISO language code, but can be any text string that you require.

Mining for Concepts and Categories

The Text Mining modeling node is used to generate one of two text mining model nuggets:

- *Concept model nuggets* uncover and extract salient concepts from your structured or unstructured text data.
- *Category model nuggets* score and assign documents and records to categories, which are made up of the extracted concepts (and patterns).

The extracted concepts and patterns as well as the categories from your model nuggets can all be combined with existing structured data, such as demographics, and applied using the full suite of tools from IBM® SPSS® Modeler to yield better and more focused decisions. For example, if customers frequently list login issues as the primary impediment to completing online account management tasks, you might want to incorporate “login issues” into your models.

Additionally, the Text Mining modeling node is fully integrated within IBM SPSS Modeler so that you can deploy text mining streams via IBM SPSS Modeler Solution Publisher for real-time scoring of unstructured data in applications such as PredictiveCallCenter. The ability to deploy these streams ensures successful closed-loop text mining implementations. For example, your organization can now analyze scratch-pad notes from inbound or outbound callers by applying your predictive models to increase the accuracy of your marketing message in real time. Using text mining model results in streams has been shown to improve the accuracy of predictive data models.

To run IBM SPSS Modeler Text Analytics with IBM SPSS Modeler Solution Publisher, add the directory `<install_directory>/ext/bin/spss.TMWBServer` to the `$LD_LIBRARY_PATH` environment variable.

In IBM SPSS Modeler Text Analytics, we often refer to extracted concepts and categories. It is important to understand the meaning of concepts and categories since they can help you make more informed decisions during your exploratory work and model building.

Concepts and Concept Model Nuggets

During the extraction process, the text data is scanned and analyzed in order to identify interesting or relevant single words, such as `election` or `peace`, and word phrases, such as `presidential election`,

election of the president, or peace treaties. These words and phrases are collectively referred to as *terms*. Using the linguistic resources, the relevant terms are extracted, and similar terms are grouped together under a lead term called a **concept**.

In this way, a concept could represent multiple underlying terms depending on your text and the set of linguistic resources you are using. For example, let's say we have a employee satisfaction survey and the concept **salary** was extracted. Let's also say that when you looked at the records associated with **salary**, you noticed that **salary** isn't always present in the text but instead certain records contained something similar, such as the terms **wage**, **wages**, and **salaries**. These terms are grouped under **salary** since the extraction engine deemed them as similar or determined they were synonyms based on processing rules or linguistic resources. In this case, any documents or records containing any of those terms would be treated as if they contained the word **salary**.

If you want to see what terms are grouped under a concept, you can explore the concept within an interactive workbench or look at which synonyms are shown in the concept model. See the topic [Underlying Terms in Concept Models](#) for more information.

A **concept model nugget** contains a set of concepts that can be used to identify records or documents that also contain the concept (including any of its synonyms or grouped terms). A concept model can be used in two ways. The first would be to explore and analyze the concepts that were discovered in the original source text or to quickly identify documents of interest. The second would be to apply this model to new text records or documents to quickly identify the same key concepts in the new documents/records, such as the real-time discovery of key concepts in scratch-pad data from a call center.

See the topic [Text Mining Nugget: Concept Model](#) for more information.

Categories and Category Model Nuggets

You can create **categories** that represent, in essence, higher-level concepts or topics to capture the key ideas, knowledge, and attitudes expressed in the text. Categories are made up of set of descriptors, such as *concepts*, *types*, and *rules*. Together, these descriptors are used to identify whether or not a record or document belongs in a given category. A document or record can be scanned to see whether any of its text matches a descriptor. If a match is found, the document/record is assigned to that category. This process is called **categorization**.

Categories can be built automatically using the product's robust set of automated techniques, manually using additional insight you may have regarding the data, or a combination of both. You can also load a set of prebuilt categories from a text analysis package through the Model tab of this node. Manual creation of categories or refining categories can only be done through the interactive workbench. See the topic [Text Mining Node: Model Tab](#) for more information.

A **category model nugget** contains a set of categories along with its descriptors. The model can be used to categorize a set of documents or records based on the text in each document/record. Every document or record is read and then assigned to each category for which a descriptor match was found. In this way, a document or record could be assigned to more than one category. You can use category model nuggets to see the essential ideas in open-ended survey responses or in a set of blog entries, for example.

See the topic [Text Mining Nugget: Category Model](#) for more information.

- [Text Mining modeling node](#)
- [Text Mining Node: Fields Tab](#)
- [Document Settings for Fields Tab](#)
- [Text Mining Node: Model Tab](#)
- [Build Interactively](#)
- [Generate Directly](#)
- [Copying resources from templates and TAPs](#)
- [Text Mining node: Expert tab](#)
- [Sampling Upstream to Save Time](#)
- [Using the Text Mining node in a stream](#)
- [Node Properties for Scripting](#)
- [Text Mining node: TextMiningWorkbench](#)
- [Text Mining Nugget: Concept Model](#)
- [Options for Including Concepts for Scoring](#)
- [Underlying Terms in Concept Models](#)
- [Text Mining Nugget: Category Model](#)
- [Text Mining model nugget: TMWBModelApplier](#)
- [Microsoft Internet Explorer settings for Help](#)

Related information

- [Text Mining modeling node](#)
- [Text Mining Node: Fields Tab](#)
- [Document Settings for Fields Tab](#)
- [Text Mining Node: Model Tab](#)
- [Build Interactively](#)
- [Generate Directly](#)
- [Copying resources from templates and TAPs](#)
- [Text Mining node: Expert tab](#)
- [Sampling Upstream to Save Time](#)
- [Using the Text Mining node in a stream](#)
- [Node Properties for Scripting](#)
- [Text Mining node: TextMiningWorkbench](#)
- [Text Mining Nugget: Concept Model](#)
- [Options for Including Concepts for Scoring](#)
- [Underlying Terms in Concept Models](#)
- [Text Mining Nugget: Category Model](#)
- [Text Mining model nugget: TMWBModelApplier](#)
- [Microsoft Internet Explorer settings for Help](#)

Text Mining modeling node

The Text Mining node uses linguistic and frequency techniques to extract key concepts from the text and create categories with these concepts and other data. The node can be used to explore the text data contents or to produce either a concept model nugget or category model nugget. When you execute this modeling node, an internal linguistic extraction engine extracts and organizes the concepts, patterns, and/or categories using natural language processing methods.

You can execute the Text Mining node and automatically produce a concept or category model nugget using the Generate directly option. Alternatively, you can use a more hands-on, exploratory approach using the Build interactively mode in which not only can you extract concepts, create categories, and refine your linguistic resources, but also perform text link analysis and explore clusters. See the topic [Text Mining Node: Model Tab](#) for more information.

You can find this node on the IBM® SPSS® Modeler Text Analytics tab of nodes palette at the bottom of the IBM SPSS Modeler window. See the topic [IBM SPSS Modeler Text Analytics nodes](#) for more information.

Requirements. Text Mining modeling nodes accept text data from a Web Feed node, File List node, or any of the standard source nodes. This node is installed with IBM SPSS Modeler Text Analytics and can be accessed on the IBM SPSS Modeler Text Analytics palette.

Note: This node replaces the Text Extraction node, which was offered in old versions of the product. If you have older streams that use the old nodes or model nuggets, you must rebuild your streams using the Text Mining node.

- [Text Mining Node: Fields Tab](#)
- [Text Mining Node: Model Tab](#)
- [Text Mining node: Expert tab](#)
- [Sampling Upstream to Save Time](#)
- [Using the Text Mining node in a stream](#)

Text Mining Node: Fields Tab

Use the Fields tab to specify the field settings for the data from which you will be extracting concepts. Consider using a Sample node upstream from this node when working with larger datasets to speed processing times. See the topic [Sampling Upstream to Save Time](#) for more information.

You can set the following parameters:

ID field Select the field containing the identifier for the text records. Identifiers must be integers. The ID field serves as an index for the individual text records. Use an ID field if the text field represents the text to be mined.

Text field. Select the field containing the text to be mined. This field depends on the data source.

Language field Select the field that contains the two letter ISO language identifier. If you do not select a field, the language of each document is assumed to be that of the supplied template.

Document type. The document type specifies the structure of the text. Select one of the following types:

- Full text. Use for most documents or text sources. The entire set of text is scanned for extraction. Unlike the other options, there are no additional settings for this option.
- Structured text. Use for bibliographic forms, patents, and any files that contain regular structures that can be identified and analyzed. This document type is used to skip all or part of the extraction process. It allows you to define term separators, assign types, and impose a minimum frequency value. If you select this option, you must click the Settings button and enter text separators in the Structured Text Formatting area of the Document Settings dialog box. See the topic [Document Settings for Fields Tab](#) for more information.

Textual unity. Select the extraction mode from the following:

- Document mode. Use for documents that are short and semantically homogenous, such as articles from news agencies.
- Paragraph mode. Use for Web pages and nontagged documents. The extraction process semantically divides the documents, taking advantage of characteristics such as internal tags and syntax. If this mode is selected, scoring is applied paragraph by paragraph.

Therefore, for example, the rule **apple & orange** is true only if **apple** and **orange** are found in the same paragraph.

Note: Due to the way text is extracted from PDF documents, Paragraph mode does not work on these documents. This is because the extraction suppresses the carriage return marker.

Paragraph mode settings. This option is available only if you set the textual unity option to Paragraph mode. Specify the character thresholds to be used in any extraction. The actual size is rounded up or down to the nearest period. To ensure that the word associations produced from the text of the document collection are representative, avoid specifying an extraction size that is too small.

- Minimum. Specify the minimum number of characters to be used in any extraction.
- Maximum. Specify the maximum number of characters to be used in any extraction.

Partition mode Use the partition mode to choose whether to partition based on the type node settings or to select another partition. Partitioning separates the data into training and test samples.

- [Document Settings for Fields Tab](#)

Related information

- [Mining for Concepts and Categories](#)
 - [Text Mining modeling node](#)
 - [Document Settings for Fields Tab](#)
 - [Text Mining Node: Model Tab](#)
 - [Build Interactively](#)
 - [Generate Directly](#)
 - [Copying resources from templates and TAPs](#)
 - [Text Mining node: Expert tab](#)
 - [Sampling Upstream to Save Time](#)
 - [Using the Text Mining node in a stream](#)
 - [Node Properties for Scripting](#)
 - [Text Mining node: TextMiningWorkbench](#)
-

Document Settings for Fields Tab

Structured Text Formatting

If you want to skip all or part of the extraction process because you have structured data or want to impose rules on how to handle the text, use the Structured text document type option and declare the fields or tags containing the text in the Structured Text Formatting section of the Document Settings dialog box. Extracted terms are derived only from the text contained within the declared fields or tags (and child tags). Any undeclared field or tag will be ignored.

In certain contexts, linguistic processing is not required, and the linguistic extraction engine can be replaced by explicit declarations. In a bibliography file where keyword fields are separated by separators such as a semicolon (;) or comma (,), it is sufficient to extract the string between two separators. For this reason, you can suspend the full extraction process and instead define special handling rules to declare term separators, assign types to the extracted text, or impose a minimum frequency count for extraction.

Use the following rules when declaring structured text elements:

- Only one field, tag, or element per line can be declared. They do not have to be present in the data.
- Declarations are case sensitive.
- If declaring a tag that has attributes, such as `<title id="_home_markdown_jenkins_workspace_Transform_in_SS3RA7_18.4.0_ta_guide_ddita_textmining_tmfc_textmining_fields_docsettings_1234">`, and you want to include all variations or, in this case, all IDs, add the tag without the attribute or the ending angle bracket (>), such as `<title`
- Add a colon after the field or tag name to indicate that this is structured text. Add this colon directly after the field or tag but before any separators, types, or frequency values, such as `author:` or `<place>:` .
- To indicate that multiple terms are contained in the field or tag and that a separator is being used to designate the individual terms, declare the separator after the colon, such as `author: ,` or `<section>: ;` .
- To assign a type to the content found in the tag, declare the type name after the colon and a separator, such as `author: ,Person` or `<place>: ;Location` . Declare type using the names as they appear in the Resource Editor.
- To define a minimum frequency count for a field or tag, declare a number at the end of the line, such as `author: ,Person1` or `<place>: ;Location5` . Where `n` is the frequency count you defined, terms found in the field or tag must occur at least `n` times in the entire set of documents or records to be extracted. This also requires you to define a separator.
- If you have a tag that contains a colon, you must precede the colon with a backslash character so that the declaration is not ignored. For example, if you have a field called `<topic:source>`, enter it as `<topic\ :source>`.

To illustrate the syntax, let's assume you have the following recurring bibliographic fields:

```
author:Morel, Kawashima
abstract:This article describes how fields are declared.
publication:Text Mining Documentation
datepub:March 2010
```

For this example, if we wanted the extraction process to focus on author and abstract but ignore the rest of the content, we would declare only the following fields:

```
author: ,Person1
abstract:
```

In this example, the `author: ,Person1` field declaration states that linguistic processing was suspended on the field contents. Instead, it states that the author field contains more than one name, which is separated from the next by a comma separator, and these names should be assigned to the Person type and that if the name occurs at least once in the entire set of documents or records, it should be extracted. Since the field `abstract:` is listed without any other declarations, the field will be scanned during extraction and standard linguistic processing and typing will be applied.

XML Text Formatting

If you want to limit the extraction process to only the text within specific XML tags, use the XML text document type option and declare the tags containing the text in the XML Text Formatting section of the Document Settings dialog box. Extracted terms are derived only from the text contained within these tags or their child tags.

Important! If you want to skip the extraction process and impose rules on term separators, assign types to the extracted text, or impose a frequency count for extracted terms, use the Structured text option described next.

Use the following rules when declaring tags for XML text formatting:

- Only one XML tag per line can be declared.
- Tag elements are case sensitive.
- If a tag has attributes, such as `<title id="_home_markdown_jenkins_workspace_Transform_in_SS3RA7_18.4.0_ta_guide_ddita_textmining_tmfc_textmining_fields_docsettings_1234">`, and you want to include all variations or, in this case, all IDs, add the tag without the attribute or the ending angle bracket (>), such as `<title`

To illustrate the syntax, let's assume you have the following XML document:

```
<section>Rules of the Road
  <title
id="_home_markdown_jenkins_workspace_Transform_in_SS3RA7_18.4.0_ta_guide_ddita_textmining_tmfc_textmining_fields_docsettings_01234">Traffic Signals</title>
    <p>Road signs are helpful.</p>
  </section>
  <p>Learning the rules is important.</p>
```

For this example, we will declare the following tags:

```
<section>
<title
```

In this example, since you have declared the tag `<section>`, the text in this tag and its nested tags, `Traffic Signals` and `Road signs are helpful`, are scanned during the extraction process. However, `Learning the rules is important` is ignored since the tag `<p>` was not explicitly declared nor was the tag nested within a declared tag.

Related information

- [Mining for Concepts and Categories](#)
 - [Text Mining modeling node](#)
 - [Text Mining Node: Fields Tab](#)
 - [Text Mining Node: Model Tab](#)
 - [Build Interactively](#)
 - [Generate Directly](#)
 - [Copying resources from templates and TAPs](#)
 - [Text Mining node: Expert tab](#)
 - [Sampling Upstream to Save Time](#)
 - [Using the Text Mining node in a stream](#)
 - [Node Properties for Scripting](#)
 - [Text Mining node: TextMiningWorkbench](#)
-

Text Mining Node: Model Tab

Use the Model tab to specify the build method and general model settings for the node output.

You can set the following parameters:

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Build mode. Specifies how the model nuggets will be produced when a stream with this Text Mining node runs. Alternatively, you can use a more hands-on, exploratory approach using the Build interactively mode in which not only can you extract concepts, create categories, and refine your linguistic resources but you can also perform text link analysis and explore clusters.

- Build interactively. When a stream runs, this option launches an interactive interface in which you can extract concepts and patterns, explore and fine-tune the extracted results, build and refine categories, fine-tune the linguistic resources (templates, synonyms, types, libraries, etc.), and build category model nuggets. See [Build Interactively](#) for more information.
- Generate directly. This option indicates that, when the stream runs, a model should automatically be created and added to the Models palette. Unlike the interactive workbench, no additional manipulation is needed from you at run time aside from the settings defined in the

node. If you select this option, model specific options appear with which you can define the type of model you want to produce. See [Generate Directly](#) for more information.

Store large models in AS. If you have a connection to IBM® SPSS® Analytic Server, select this option to store your models remotely on the server.

Note: Any model that is built and stored on a server can only be scored on that server. To resume an interactive workbench session that contains such a model, you need a connection to the original server that was used to create the session.

Copy resources from. When mining text, the extraction is based not only on the settings in the Expert tab but also on the linguistic resources. These resources serve as the basis for how to handle and process the text during extraction to get the concepts, types, and sometimes patterns. You can copy resources into this node from a resource template, a text analysis package (.tap), or an SPSS Text Analytics for Surveys project file (.tas). Make your selection and then click Load to define the template, package, or project from which the resources will be copied. At the moment that you load, a copy of the resources is stored in the node. Therefore, if you ever want to use an updated resource, you must reload it here or in an interactive workbench session. For your convenience, the date and time at which the resources were copied and loaded is shown in the node. See [Copying resources from templates and TAPs](#) for more information.

Text language. Identifies the language of the text being mined. The resources copied in the node control the language options presented. Select the language for which the resources were tuned.

- [Build Interactively](#)
 - [Generate Directly](#)
 - [Copying resources from templates and TAPs](#)
-

Build Interactively

In the Model tab of the text mining modeling node, you can choose a build mode for your model nuggets. If you choose Build interactively, then an interactive interface opens when you execute the stream. In this interactive workbench, you can:

- Extract and explore the extraction results, including concepts and typing to discover the salient ideas in your text data.
- Use a variety of methods to build and extend categories from concepts, types, TLA patterns, and rules so you can score your documents and records into these categories.
- Refine your linguistic resources (resource templates, libraries, dictionaries, synonyms, and more) so you can improve your results through an iterative process in which concepts are extracted, examined, and refined.
- Perform text link analysis (TLA) and use the TLA patterns discovered to build better category model nuggets. The Text Link Analysis node doesn't offer the same exploratory options or modeling capabilities.
- Generate clusters to discover new relationships and explore relationships between concept, types, patterns, and categories in the Visualization pane.
- Generate refined category model nuggets to the Models palette in IBM® SPSS® Modeler and use them in other streams.

Note: You cannot build an interactive model if you are creating an IBM SPSS Collaboration and Deployment Services job.

Use session work (categories, TLA, resources, etc.) from last node update. When you work in an interactive workbench session, you can update the node with session data (extraction parameters, resources, category definitions, etc.). The Use session work option allows you to relaunch the interactive workbench using the saved session data. This option is disabled the first time you use this node, since no session data could have been saved. To learn how to update the node with session data so that you can use this option, see [Updating Modeling Nodes and Saving](#).

If you launch a session *with* this option, then the extraction settings, categories, resources, and any other work from the last time you performed a node update from an interactive workbench session are available when you next launch a session. Since saved session data are used with this option, certain content, such as the resources copied from the template below, and other tabs are disabled and ignored. But if you launch a session *without* this option, only the contents of the node as they are defined now are used, meaning that any previous work you've performed in the workbench will not be available.

Note: If you change the source node for your stream after extraction results have been cached with the Use session work... option, you will need to run a new extraction once the interactive workbench session is launched if you want to get updated extraction results.

Skip extraction and reuse cached data and results. You can reuse any cached extraction results and data in the interactive workbench session. This option is particularly useful when you want to save time and reuse extraction results rather than waiting for a completely new extraction to be performed when the session is launched. In order to use this option, you must have previously updated this node from within an interactive workbench session and chosen the option to Keep the session work and cache text data with extraction results for reuse. To learn how to update the node with session data so that you can use this option, see [Updating Modeling Nodes and Saving](#).

Begin session by. Select the option indicating the view and action you want to take place first upon launching the interactive workbench session. Regardless of the view you start in, you can switch to any view once in the session.

- **Using extraction results to build categories.** This option launches the interactive workbench in the Categories and Concepts view and, if applicable, performs an extraction. In this view, you can create categories and generate a category model. You can also switch to another view. See the topic [Interactive workbench mode](#) for more information.
- **Exploring text link analysis (TLA) results.** This option launches and begins by extracting and identifying relationships between concepts within the text, such as opinions or other links in the Text Link Analysis view. You must select a template or text analysis package that

contains TLA pattern rules in order to use this option and obtain results. If you are working with larger datasets, the TLA extraction can take some time. In this case, you may want to consider using a Sample node upstream. See the topic [Exploring Text Link Analysis](#) for more information.

- **Analyzing co-word clusters.** This option launches in the Clusters view and updates any outdated extraction results. In this view, you can perform co-word cluster analysis, which produces a set of clusters. Co-word clustering is a process that begins by assessing the strength of the link value between two concepts based on their co-occurrence in a given record or document and ends with the grouping of strongly linked concepts into clusters. See the topic [Interactive workbench mode](#) for more information.

Related information

- [Mining for Concepts and Categories](#)
 - [Text Mining modeling node](#)
 - [Text Mining Node: Fields Tab](#)
 - [Document Settings for Fields Tab](#)
 - [Text Mining Node: Model Tab](#)
 - [Generate Directly](#)
 - [Copying resources from templates and TAPs](#)
 - [Text Mining node: Expert tab](#)
 - [Sampling Upstream to Save Time](#)
 - [Using the Text Mining node in a stream](#)
 - [Node Properties for Scripting](#)
 - [Text Mining node: TextMiningWorkbench](#)
-

Generate Directly

In the Model tab of the text mining modeling node, you can choose a build mode for your model nuggets. If you choose Generate directly, you can set the options in the node and then just execute your stream. The output is a concept model nugget, which was placed directly in the Models palette. Unlike the interactive workbench, no additional manipulation is needed from you at execution time besides the frequency settings defined for this option in the node.

Maximum number of concepts to include in model. This option, which applies only when you build a model automatically (non-interactive), indicates that you want to create a concept model. It also states that this model should contain no more than the specified number of concepts.

- **Check concepts based on highest frequency. Top number of concepts.** Starting with the concept with the highest frequency, this is the number of concepts that will be checked. Here, frequency refers to the number of times a concept (and all its underlying terms) appears in the entire set of the documents/records. This number could be higher than the record count, since a concept can appear multiple times in a record.
- **Uncheck concepts that occur in too many records. Percentage of records.** Unchecks concepts with a record count percentage higher than the number you specified. This option is useful for excluding concepts that occur frequently in your text or in every record but have no significance in your analysis.

Optimize for speed of scoring. Selected by default, this option ensures that the model created is compact and scores at high speed. Deselecting this option creates a much larger model which scores more slowly. However, the larger model ensures that scores displayed initially in the generated concept model are the same as those obtained when scoring the same text with the model nugget.

Related information

- [Mining for Concepts and Categories](#)
 - [Text Mining modeling node](#)
 - [Text Mining Node: Fields Tab](#)
 - [Document Settings for Fields Tab](#)
 - [Text Mining Node: Model Tab](#)
 - [Build Interactively](#)
 - [Copying resources from templates and TAPs](#)
 - [Text Mining node: Expert tab](#)
 - [Sampling Upstream to Save Time](#)
 - [Using the Text Mining node in a stream](#)
 - [Node Properties for Scripting](#)
 - [Text Mining node: TextMiningWorkbench](#)
-

Copying resources from templates and TAPs

When mining text, the extraction is based not only on the settings in the Expert tab but also on the linguistic resources. These resources serve as the basis for how to handle and process the text during extraction in order to get the concepts, types, and sometimes patterns. You can copy resources into this node from a *resource template*, and if you are in the Text Mining node, you can also select a *text analysis package* (TAP) or an SPSS® Text Analytics for Surveys project (.tas).

By default, resources are copied into the node from the basic template for licensed languages for your product when you add the node to the canvas. If you have licenses for multiple languages, the first language selected is used to determine the template to load automatically.

At the moment that you load, a copy of the selected resources is stored in the node. Only the contents of the template, TAP, or SPSS Text Analytics for Surveys project resources are copied while the template, TAP, or SPSS Text Analytics for Surveys itself is not linked to the node. This means that if resources are later updated, these updates are not automatically available in the node. In short, the resources loaded into the node are always used unless you either reload a new copy of the resources, or unless you update a Text Mining node and select the Use session work option. For more information on Use session work, see further in this section.

When you select a resource, choose one with the same language as your text data. You can only use resources in the languages for which you are licensed. If you want to perform text link analysis, you must select a template that contains TLA patterns. If a template contains TLA patterns, an icon will appear in the TLA column of the Load Resource Template dialog box.

Note: You cannot load TAPs or SPSS Text Analytics for Surveys projects into the Text Link Analysis node.

Resource templates

A resource template is a predefined set of libraries and advanced linguistic and nonlinguistic resources that have been fine-tuned for a particular domain or usage. In the text mining modeling node, a copy of the resources from a basic template are already loaded in the node when you add the node to the stream, but you can change templates or load a text analysis package by selecting either Resource template or Text analysis package and then clicking Load. For templates, you can then select the template in the Load Resource Template dialog box.

Note: If you do not see the template you want in the list but you have an exported copy on your machine, you can import it now. You can also export from this dialog box to share with other users. See [Importing and Exporting Templates](#) for more information.

Text analysis packages (TAPs) and Text Analysis for Surveys projects (TAS)

A text analysis package (TAP) is a predefined set of libraries and advanced linguistic and nonlinguistic resources bundled with one or more sets of predefined categories. IBM® SPSS Modeler Text Analytics offers several prebuilt TAPs, which is fine-tuned for a specific domain. You can edit these TAPs and save them to a different directory to use them to jump start your category model building. You can also create your own TAPs in the interactive session. See [Loading Text Analysis Packages](#) for more information.

If you choose to import a SPSS Text Analytics for Surveys project (.tas), it will be converted to a TAP.

Note: You cannot load TAPs or SPSS Text Analytics for Surveys projects into the Text Link Analysis node.

Using the "Use Session Work" option (Model tab)

While resources are copied into the node in the Model tab, you might also make changes later to the resources in an interactive session and want to update the text mining modeling node with these latest changes. In this case, you would select the Use session work option in the Model tab of the text mining modeling node.

If you select Use session work, the Load button is disabled in the node to indicate that those resources that came from the interactive workbench will be used instead of the resources that were loaded here previously.

To make changes to resources once you've selected the Use session work option, you can edit or switch your resources directly inside the interactive workbench session through the Resource Editor view. See [Updating Node Resources After Loading](#) for more information.

Text Mining node: Expert tab

The Expert tab contains certain advanced parameters that impact how text is extracted and handled. The parameters in this dialog box control the basic behavior, as well as a few advanced behaviors, of the extraction process. However, they represent only a portion of the options available to you. There are also a number of linguistic resources and options that impact the extraction results, which are controlled by the resource template you select on the Model tab. See the topic [Text Mining Node: Model Tab](#) for more information.

Note: This entire tab is disabled if you have selected the Build interactively mode using saved interactive workbench information on the Model tab, in which case the extraction settings are taken from the last saved workbench session.

You can set the following parameters whenever extracting:

Limit extraction to concepts with a global frequency of at least [n]. Specifies the minimum number of times a word or phrase must occur in the text in order for it to be extracted. In this way, a value of 5 limits the extraction to those words or phrases that occur at least five times in the entire set of records or documents.

In some cases, changing this limit can make a big difference in the resulting extraction results, and consequently, your categories. Let's say that you are working with some restaurant data and you do not increase the limit above 1 for this option. In this case, you might find *pizza* (1), *thin pizza* (2), *spinach pizza* (2), and *favorite pizza* (2) in your extraction results. However, if you were to limit the extraction to a global frequency of 5 or more and re-extract, you would no longer get three of these concepts. Instead you would get *pizza* (7), since *pizza* is the simplest form and also this word already existed as a possible candidate. And depending on the rest of your text, you might actually have a frequency of more than seven, depending on whether there are still other phrases with *pizza* in the text. Additionally, if *spinach pizza* was already a category descriptor, you might need to add *pizza* as a descriptor instead to capture all of the records. For this reason, change this limit with care whenever categories have already been created.

Note that this is an extraction-only feature; if your template contains terms (which they usually do), and a term for the template is found in the text, then the term will be indexed regardless of its frequency.

For example, suppose you use a Basic Resources template that includes "los angeles" under the <Location> type in the Core library; if your document contains Los Angeles only once, then Los Angeles will be part of the list of concepts. To prevent this you will need to set a filter to display concepts occurring at least the same number of times as the value entered in the Limit extraction to concepts with a global frequency of at least [n] field.

Accommodate punctuation errors. This option temporarily normalizes text containing punctuation errors (for example, improper usage) during extraction to improve the extractability of concepts. This option is extremely useful when text is short and of poor quality (as, for example, in open-ended survey responses, e-mail, and CRM data), or when the text contains many abbreviations.

Accommodate spelling for a minimum word character length of [n] This option applies a fuzzy grouping technique that helps group commonly misspelled words or closely spelled words under one concept. The fuzzy grouping algorithm temporarily strips all vowels (except the first one) and strips double/triple consonants from extracted words and then compares them to see if they are the same so that *modeling* and *modelling* would be grouped together. However, if each term is assigned to a different type, excluding the <Unknown> type, the fuzzy grouping technique will not be applied.

You can also define the minimum number of root characters required before fuzzy grouping is used. The number of root characters in a term is calculated by totaling all of the characters and subtracting any characters that form inflection suffixes and, in the case of compound-word terms, determiners and prepositions. For example, the term *exercises* would be counted as 8 root characters in the form "exercise," since the letter s at the end of the word is an inflection (plural form). Similarly, *apple sauce* counts as 10 root characters ("apple sauce") and *manufacturing of cars* counts as 16 root characters ("manufacturing car"). This method of counting is only used to check whether the fuzzy grouping should be applied but does not influence how the words are matched.

Note: If you find that certain words are later grouped incorrectly, you can exclude word pairs from this technique by explicitly declaring them in the Fuzzy Grouping: Exceptions section in the Advanced Resources tab. See the topic [Fuzzy Grouping](#) for more information.

Extract uniterms This option extracts single words (uniterms) as long as the word is not already part of a compound word and if it is either a noun or an unrecognized part of speech.

Extract nonlinguistic entities This option extracts nonlinguistic entities, such as phone numbers, social security numbers, times, dates, currencies, digits, percentages, e-mail addresses, and HTTP addresses. You can include or exclude certain types of nonlinguistic entities in the Nonlinguistic Entities: Configuration section of the Advanced Resources tab. By disabling any unnecessary entities, the extraction engine won't waste processing time. See the topic [Configuration](#) for more information.

Uppercase algorithm This option extracts simple and compound terms that are not in the built-in dictionaries as long as the first letter of the term is in uppercase. This option offers a good way to extract most proper nouns.

Group partial and full person names together when possible This option groups names that appear differently in the text together. This feature is helpful since names are often referred to in their full form at the beginning of the text and then only by a shorter version. This option attempts to match any uniterm with the <Unknown> type to the last word of any of the compound terms that is typed as <Person>. For example, if *doe* is found and initially typed as <Unknown>, the extraction engine checks to see if any compound terms in the <Person> type include *doe* as the last word, such as *john doe*. This option does not apply to first names since most are never extracted as uniterms.

Maximum nonfunction word permutation This option specifies the maximum number of nonfunction words that can be present when applying the permutation technique. This permutation technique groups similar phrases that differ from each other only by the nonfunction words (for example, *of* and *the*) contained, regardless of inflection. For example, let's say that you set this value to at most two words and both *company officials* and *officials of the company* were extracted. In this case, both extracted terms would be grouped together in the final concept list since both terms are deemed to be the same when *of the* is ignored.

Use derivation when grouping multiterms When processing Big Data, select this option to group multiterms by using derivation rules.

Note: To enable the extraction of Text Link Analysis results, you must begin the session with the Exploring text link analysis results option and also choose resources that contain TLA definitions. You can always extract TLA results later during an interactive workbench session through the Extraction Settings dialog. See the topic [Extracting data](#) for more information.

Sampling Upstream to Save Time

When you have a large amount of data, the processing times can take minutes to hours, especially when using the interactive workbench session. The greater the size of the data, the more time the extraction and categorization processes will take. To work more efficiently, you can add aIBM®

SPSS® Modeler Sample nodes upstream from your Text Mining node. Use this Sample node to take a random sample using a smaller subset of documents or records to do the first few passes.

A smaller sample is often perfectly adequate to decide how to edit your resources and even create most if not all of your categories. And once you have run on the smaller dataset and are satisfied with the results, you can apply the same technique for creating categories to the entire set of data. Then you can look for documents or records that do not fit the categories you have created and make adjustments as needed.

Note: The Sample node is a standard IBM SPSS Modeler node.

Related information

- [Mining for Concepts and Categories](#)
- [Text Mining modeling node](#)
- [Text Mining Node: Fields Tab](#)
- [Document Settings for Fields Tab](#)
- [Text Mining Node: Model Tab](#)
- [Build Interactively](#)
- [Generate Directly](#)
- [Copying resources from templates and TAPs](#)
- [Text Mining node: Expert tab](#)
- [Using the Text Mining node in a stream](#)
- [Node Properties for Scripting](#)
- [Text Mining node: TextMiningWorkbench](#)
- [Text Link Analysis node](#)
- [Text Link Analysis node: Fields tab](#)
- [Text Link Analysis Node: Model Tab](#)
- [Text Link Analysis node: Expert tab](#)
- [TLA node output](#)
- [Caching TLA Results](#)
- [Using the Text Link Analysis node in a stream](#)
- [Text Link Analysis node: textlinkanalysis](#)
- [Microsoft Internet Explorer settings for Help](#)

Using the Text Mining node in a stream

The Text Mining modeling node is used to access data and extract concepts in a stream. You can use any source node to access data, such as a Database node, Var. File node, Web Feed node, or Fixed File node. For text that resides in external documents, a File List node can be used.

Example 1: File List node and Text Mining node to build a concept model nugget directly

The following example shows how to use the File list node along with the Text Mining modeling node to generate the concept model nugget. For more information on using the File List node, see [File List node](#).

1. File List node (Settings tab). First, we added this node to the stream to specify where the text documents are stored. We selected the directory containing all of the documents on which we want to perform text mining.
2. Text Mining node (Fields tab). Next, we added and connected a Text Mining node to the File List node. In this node, we defined our input format, resource template, and output format. We selected the field name produced from the File List node and selected the text field, as well as other settings. See the topic [Using the Text Mining node in a stream](#) for more information.
3. Text Mining node (Model tab). Next, on the Model tab, we selected the build mode to generate a concept model nugget directly from this node. You can select a different resource template, or keep the basic resources.

Example 2: Excel File and Text Mining nodes to build a category model interactively

This example shows how the Text Mining node can also launch an interactive workbench session. For more information on the interactive workbench, see [Interactive workbench mode](#).

1. Excel source node (Data tab). First, we added this node to the stream to specify where the text is stored.
2. Text Mining node (Fields tab). Next, we added and connected a Text Mining node. On this first tab, we defined our input format. We selected a field name from the source node.
3. Text Mining node (Model tab). Next, on the Model tab, we selected to build a category model nugget interactively and to use the extraction results to build categories automatically. In this example, we loaded a copy of resources and a set of categories from a text analysis

package.

4. Interactive Workbench session. Next, we executed the stream, and the interactive workbench interface opened. After an extraction was performed, we began exploring our data and improving our categories.
-

Text Mining Nugget: Concept Model

A Text Mining concept model nugget is created whenever you successfully execute a Text Mining model node where you've selected the option to Generate a model directly in the Model tab. A text mining concept model nugget is used for the real-time discovery of key concepts in other text data, such as scratch-pad data from a call center.

The concept model nugget itself comprises a list of concepts, which have been assigned to types. You can select any or all of the concepts in that model for scoring against other data. When you execute a stream containing a Text Mining model nugget, new fields are added to the data according to the build mode selected on the Model tab of the Text Mining modeling node prior to building the model. See the topic [Concept Model: Model Tab](#) for more information.

If the model nugget was generated using translated documents, the scoring will be performed in the translated language. Similarly, if the model nugget was generated using English as the language, you can specify a translation language in the model nugget, since the documents will then be translated into English.

Text Mining model nuggets are placed in the model nugget palette (located on the Models tab in the upper right side of the IBM® SPSS® Modeler window) when they are generated.

Viewing Results

To see information about the model nugget, right-click the node in the model nuggets palette and choose Browse from the context menu (or Edit for nodes in a stream).

Adding Models to Streams

To add the model nugget to your stream, click the icon in the model nuggets palette and then click the stream canvas where you want to place the node. Or right-click the icon and choose Add to Stream from the context menu. Then connect your stream to the node, and you are ready to pass data to generate predictions.

Caution: If you want to use a scoring nugget to regenerate a modeling node that contains both the category model and the template used, we recommend that you create a TAP and use it in an interactive session, in place of the modeling node, before generating the scoring nugget.

- [Concept Model: Model Tab](#)
- [Concept model: Settings tab](#)
- [Concept Model: Fields tab](#)
- [Concept Model: Summary Tab](#)
- [Using Concept Model Nuggets in a Stream](#)

Related information

- [Text Mining modeling node](#)
 - [Mining for Concepts and Categories](#)
 - [Options for Including Concepts for Scoring](#)
 - [Underlying Terms in Concept Models](#)
 - [Text Mining Nugget: Category Model](#)
 - [Text Mining model nugget: TMWBModelApplier](#)
 - [Concept Model: Model Tab](#)
 - [Concept model: Settings tab](#)
 - [Concept Model: Fields tab](#)
 - [Concept Model: Summary Tab](#)
 - [Using Concept Model Nuggets in a Stream](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Concept Model: Model Tab

In concept models, the Model tab displays the set of concepts that were extracted. The concepts are presented in a table format with one row for each concept. The objective on this tab is to select which of the concepts will be used for scoring.

Note: If you generated a category model nugget instead, this tab will present different information. See the topic [Category Model Nugget: Model Tab](#) for more information.

All concepts are selected for scoring by default, as shown in the check boxes in the leftmost column. A checked box means that the concept will be used for scoring. An unchecked box means that the concept will be excluded from scoring. You can check multiple rows by selecting them and clicking one of the check boxes in your selection.

To learn more about each concept, you can look at the additional information provided in each of the following columns:

Concept. This is the lead word or phrase that was extracted. In some cases, this concept represents the concept name as well as some other underlying terms associated with this concept. To see which underlying terms are part of a concept, display the Underlying Terms pane inside this tab and select the concept to see the corresponding terms at the bottom of the dialog box. See the topic [Underlying Terms in Concept Models](#) for more information.

Global. Here, global (frequency) refers to the number of times a concept (and all its underlying terms) appears in the entire set of the documents/records.

- **Bar chart.** The global frequency of this concept in the text data presented as a bar chart. The bar takes the color of the type to which the concept is assigned in order to visually distinguish the types.
- **%.** The global frequency of this concept in the text data presented as a percentage.
- **N.** The actual number of occurrences of this concept in the text data.

Docs. Here, Docs refers to the document count, meaning number of documents or records in which the concept (and all its underlying terms) appears.

- **Bar chart.** The document count for this concept presented as a bar chart. The bar takes the color of the type to which the concept is assigned in order to visually distinguish the types.
- **%.** The document count for this concept presented as a percentage.
- **N.** The actual number of documents or records containing this concept.

Type. The type to which the concept is assigned. For each concept, the Global and Docs columns appear in a color to denote the type to which this concept is assigned. A **type** is a semantic groupings of concepts. See the topic [Type dictionaries](#) for more information.

Working with Concepts

By right-clicking a cell in the table, you can display a context menu in which you can:

- **Select All.** All rows in the table will be selected.
- **Copy.** The selected concept(s) are copied to the clipboard.
- **Copy With Fields** The selected concept(s) are copied to the clipboard along with the column heading.
- **Check Selected.** Checks all check boxes for the selected rows in the table thereby including those concepts for scoring.
- **Uncheck Selected.** Unchecks all check boxes for the selected rows in the table.
- **Check All.** Checks all check boxes in the table. This results in all concepts being used in the final output.
- **Uncheck All.** Unchecks all check boxes in the table. Unchecking a concept means that it will not be used in the final output.
- **Include Concepts.** Displays the Include Concepts dialog box. See the topic [Options for Including Concepts for Scoring](#) for more information.
- [Options for Including Concepts for Scoring](#)
- [Underlying Terms in Concept Models](#)

Related information

- [Text Mining Nugget: Concept Model](#)
- [Concept model: Settings tab](#)
- [Concept Model: Fields tab](#)
- [Concept Model: Summary Tab](#)
- [Using Concept Model Nuggets in a Stream](#)
- [Microsoft Internet Explorer settings for Help](#)

Options for Including Concepts for Scoring

To quickly check or uncheck those concepts that will be used for scoring, click the toolbar button for **Include Concepts..**

Figure 1. Include Concepts toolbar button



Clicking this toolbar button will open the Include Concepts dialog box to allow you to select concepts based on rules. All concepts that have a check mark on the Model tab will be included for scoring. Apply a rule in this subdialog to change which concepts will be used for scoring.

You can choose from the following options:

Check concepts based on highest frequency. Top number of concepts. Starting with the concept with the highest global frequency, this is the number of concepts that will be checked. Here, frequency refers to the number of times a concept (and all its underlying terms) appears in the entire set of the documents/records. This number could be higher than the record count, since a concept can appear multiple times in a record.

Check concepts based on document count. Minimum count. This is the lowest document count needed for the concepts to be checked. Here, document count refers to number of documents/records in which the concept (and all its underlying terms) appears.

Check concepts assigned to the type. Select a type from the drop-down list to check all concepts that are assigned to this type. Concepts are assigned to types automatically during the extraction process. A **type** is a semantic grouping of concepts. Types include such things as higher-level concepts, positive and negative words and qualifiers, contextual qualifiers, first names, places, organizations, and more. See the topic [Type dictionaries](#) for more information.

Uncheck concepts that occur in too many records. Percentage of records. Unchecks concepts with a record count percentage higher than the number you specified. This option is useful for excluding concepts that occur frequently in your text or in every record but have no significance in your analysis.

Uncheck concepts assigned to the type. Unchecks concepts matching the type that you select from the drop-down list.

Related information

- [Mining for Concepts and Categories](#)
 - [Text Mining Nugget: Concept Model](#)
 - [Underlying Terms in Concept Models](#)
 - [Text Mining Nugget: Category Model](#)
 - [Text Mining model nugget: TMWBModelApplier](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Underlying Terms in Concept Models

You can see the underlying terms that are defined for the concepts that you have selected in the table. By clicking the underlying terms toggle button on the toolbar, you can display the underlying terms table in a split pane at the bottom of the dialog.

These underlying terms include the synonyms defined in the linguistic resources (regardless of whether they were found in the text or not) as well as any extracted plural/singular forms found in the text used to generate the model nugget, permuted terms, terms from fuzzy grouping, and so on.

Figure 1. Display Underlying Terms toolbar button



Note: You cannot edit the list of underlying terms. This list is generated through substitutions, synonym definitions (in the substitution dictionary), fuzzy grouping, and more—all of which are defined in the linguistic resources. In order to make changes to how terms are grouped under a concept or how they are handled, you must make changes directly in the resources (editable in the Resource Editor in the interactive workbench or in the Template Editor and then reload in the node) and then reexecute the stream to get a new model nugget with the updated results.

By right-clicking the cell containing an underlying term or concept, you can display a context menu in which you can:

- **Copy.** The selected cell is copied to the clipboard.
- **Copy With Fields.** The selected cell is copied to the clipboard along with the column headings.
- **Select All.** All cells in the table will be selected.

Related information

- [Mining for Concepts and Categories](#)
 - [Text Mining Nugget: Concept Model](#)
 - [Options for Including Concepts for Scoring](#)
 - [Text Mining Nugget: Category Model](#)
 - [Text Mining model nugget: TMWBModelApplier](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Concept model: Settings tab

The Settings tab is used to define the text field value for the new input data, if necessary. It is also the place where you define the data model for your output (scoring mode).

Note: This tab appears only when the model nugget is placed onto the canvas. It does not exist when you are accessing this dialog box directly in the Models palette.

Scoring mode: Concepts as records

With this scoring mode, a new record is created for each **concept/document** pair. Typically, there are more records in the output than there were in the input.

In addition to the input fields, the following new fields are added to the data:

Table 1. Output fields for "Concepts as records"

Field	Description
Concept	Contains the extracted concept name found in the text data field.
Type	Stores the type of the concept as a full type name, such as <i>Location</i> or <i>Person</i> . A type is a semantic grouping of concepts. See the topic Type dictionaries for more information.
Count	Displays the number of occurrences for that concept (and its underlying terms) in the text body (record/document).

When you select this option, all other options except Accommodate punctuation errors are disabled.

Scoring mode: Concepts as fields

In concept models, for each input record, a new record is created for every concept found in a given document. Therefore, there are just as many output records as there were in the input. However, each record (row) now contains one new field (column) for each concept that was selected (using the check mark) on the Model tab. The value for each concept field depends on whether you select Flags or Counts as your field value on this tab.

Note: If you are using very large data sets, for example with a Db2 database, using Concepts as fields may encounter processing problems due to the amount of data. In this case we recommend using Concepts as records instead.

Field Values. Choose whether the new field for each concept will contain a count or a flag value.

- Flags. This option is used to obtain flags with two distinct values in the output, such as Yes/No, True/False, T/F, or 1 and 2. The storage types are set automatically to reflect the values chosen. For example, if you enter numeric values for the flags, they will be automatically handled as an integer value. The storage types for flags can be string, integer, real number, or date/time. Enter a flag value for True and for False.
- Counts. Used to obtain a count of how many times the concept occurred in a given record.

Field name extension. Specify an extension for the field name. Field names are generated by using the concept name plus this extension.

- Add as. Specify where the extension should be added to the field name. Choose Prefix to add the extension to the beginning of the string. Choose Suffix to add the extension to the end of the string.

Accommodate punctuation errors. This option temporarily normalizes text containing punctuation errors (for example, improper usage) during extraction to improve the extractability of concepts. This option is extremely useful when text is short and of poor quality (as, for example, in open-ended survey responses, e-mail, and CRM data), or when the text contains many abbreviations.

Concept Model: Fields tab

The Fields tab defines the text field value for the new input data, if necessary.

Note: This tab appears only when the model nugget is placed in the stream. It does not exist when you are accessing this output directly in the Models palette.

Text field. Select the field containing the text to be mined. This field depends on the data source.

Document type. The document type specifies the structure of the text. Select one of the following types:

- Full text. Use for most documents or text sources. The entire set of text is scanned for extraction. Unlike the other options, there are no additional settings for this option.
- Structured text. Use for bibliographic forms, patents, and any files that contain regular structures that can be identified and analyzed. This document type is used to skip all or part of the extraction process. It allows you to define term separators, assign types, and impose a minimum frequency value. If you select this option, you must click the Settings button and enter text separators in the Structured Text Formatting area of the Document Settings dialog box. See the topic [Document Settings for Fields Tab](#) for more information.

Input encoding. This option is available only if you indicated that the text field represents Pathnames to documents. It specifies the default text encoding. A conversion is done from the specified or recognized encoding to **ISO-8859-1**. So even if you specify another encoding, the extraction engine will convert it to **ISO-8859-1** before it is processed. Any characters that do not fit into the **ISO-8859-1** encoding definition will be converted to spaces.

Text language. Identifies the language of the text being mined; this is the main language detected during extraction. Contact your sales representative if you are interested in purchasing a license for a supported language for which you do not currently have access.

Concept Model: Summary Tab

The Summary tab presents information about the model itself (*Analysis* folder), fields used in the model (*Fields* folder), settings used when building the model (*Build Settings* folder), and model training (*Training Summary* folder).

When you first browse a modeling node, the folders on the Summary tab are collapsed. To see the results of interest, use the expander control to the left of the folder to show the results, or click the Expand All button to show all results. To hide the results after viewing them, use the expander control to collapse the specific folder that you want to hide, or click the Collapse All button to collapse all folders.

Related information

- [Text Mining Nugget: Concept Model](#)
- [Concept Model: Model Tab](#)
- [Concept model: Settings tab](#)
- [Concept Model: Fields tab](#)
- [Using Concept Model Nuggets in a Stream](#)
- [Text Mining Nugget: Category Model](#)
- [Category Model Nugget: Model Tab](#)
- [Category model nugget: Settings tab](#)
- [Category Model Nugget: Other Tabs](#)
- [Using category model nuggets in a stream](#)
- [Microsoft Internet Explorer settings for Help](#)

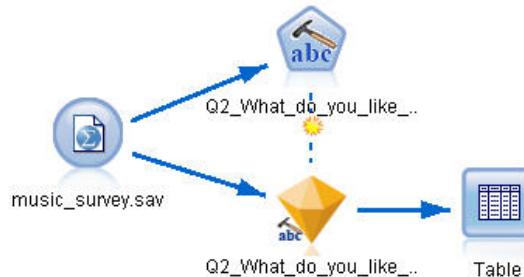
Using Concept Model Nuggets in a Stream

When using a Text Mining modeling node, you can generate either a concept model nugget or a category model nugget (through an interactive workbench session). The following example shows how to use a concept model in a simple stream.

Example: Statistics File node with the concept model nugget

The following example shows how to use the Text Mining concept model nugget.

Figure 1. Example stream: Statistics File node with a Text Mining concept model nugget



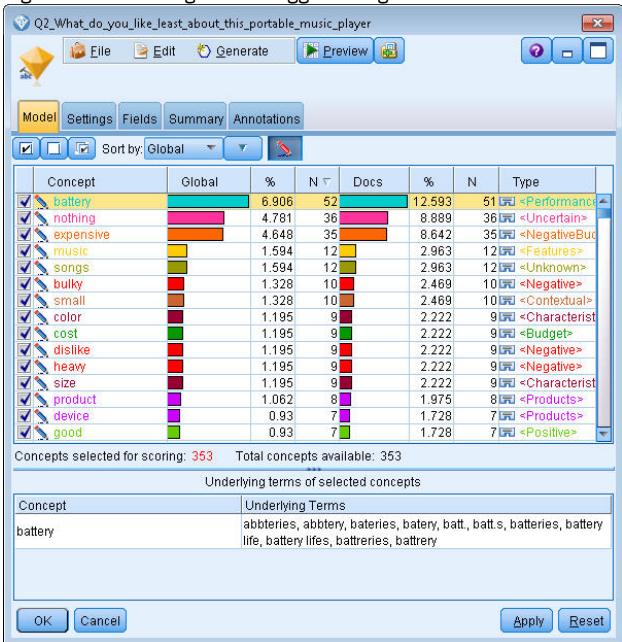
1. **Statistics File node (Data tab).** First, we added this node to the stream to specify where the text documents are stored.

Figure 2. Statistics File node dialog box: Data tab



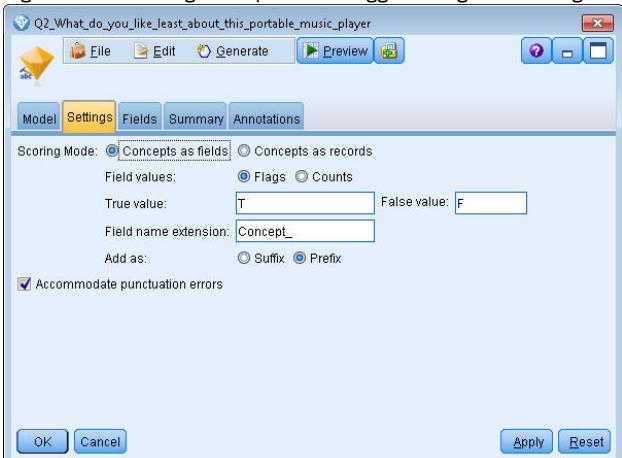
2. Text Mining concept model nugget (Model tab). Next, we added and connected a concept model nugget to the Statistics File node. We selected the concepts we wanted to use to score our data.

Figure 3. Text Mining model nugget dialog box: Model tab



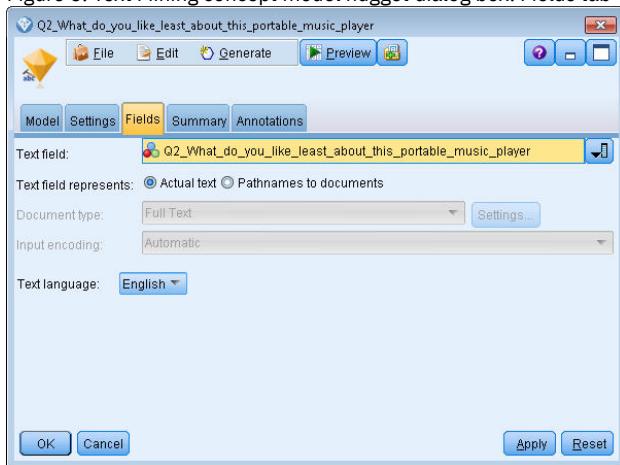
3. Text Mining concept model nugget (Settings tab). Next, we defined the output format and selected *Concepts as fields*. One new field will be created in the output for each concept selected in the Model tab. Each field name will be made up of the concept name and the prefix "Concept_".

Figure 4. Text Mining concept model nugget dialog box: Settings tab



4. Text Mining concept model nugget (Fields tab). Next, we selected the text field, Q2_What_do_you_like_least_about_this_portable_music_player, which is the field name coming from the Statistics File node. We also selected the option Text field represents: Actual text.

Figure 5. Text Mining concept model nugget dialog box: Fields tab



5. Table node. Next, we attached a table node to see the results and executed the stream. The table output opens on screen.

Figure 6. Table output scrolled to show the concept flags

	Respondent_ID	Q1_V..._Q2_What_do_you_like_least_about_this_portable_music_player	Concept_reliable	Concept_downloading...	Concept_white color	Concept_limited
1	1	little... expensive	F	F	F	F
2	2	The ba...The screen is hard to see when outside.	F	F	F	F
3	3	cost a... difficult software	F	F	F	F
4	4	Having... Nothing, I love it!	F	F	F	F
5	5	The sh...Battery life seems shorter than advertised.	F	F	F	F
6	6	Batter... Ubiquitousness; everyone has one.	F	F	F	F
7	7	I like it... I wish the 40GB model was still available. I have a 20GB model and need more memory.	F	F	F	F
8	8	portab... It doesn't have a light.	F	F	F	F
9	9	Small... Nothing. I love it.	F	F	F	F
10	10	Able... It is in the shop due to a hardware failure.	F	F	F	F
11	11	It's por... smudges on the display	F	F	F	F
12	12	Living... Battery life	F	F	F	F
13	13	mobility Technical difficulties setting it up initially and managing the library of songs on my PC.	F	F	F	F
14	14	I like th... It is a little heavy, and the battery life isn't long enough.	F	F	F	F
15	15	If hold... Battery life.	F	F	F	F
16	16	It's fun... nothing	F	F	F	F
17	17	Its cool... battery	F	F	F	F
18	18	lot of... It was very expensive	F	F	F	F
19	19	Others... I find the controls hard to use.	F	F	F	F
20	20	lightw... so small afraid I'll lose it easily	F	F	F	F

Text Mining Nugget: Category Model

A Text Mining category model nugget is created whenever you generate a category model from within the interactive workbench. This modeling nugget contains a set of categories, whose definition is made up of concepts, types, TLA patterns, and/or category rules. The nugget is used to categorize survey responses, blog entries, other Web feeds, and any other text data.

If you launch an interactive workbench session in the modeling node, you can explore the extraction results, refine the resources, fine-tune your categories before you generate category models. When you execute a stream containing a Text Mining model nugget, new fields are added to the data according to the build mode selected on the Model tab of the Text Mining modeling node prior to building the model. See the topic [Category Model Nugget: Model Tab](#) for more information.

If the model nugget was generated using translated documents, the scoring will be performed in the translated language. Similarly, if the model nugget was generated using English as the language, you can specify a translation language in the model nugget, since the documents will then be translated into English.

Text Mining model nuggets are placed in the model nugget palette (located on the Models tab in the upper right side of the IBM® SPSS® Modeler window) when they are generated.

Viewing Results

To see information about the model nugget, right-click the node in the model nuggets palette and choose Browse from the context menu (or Edit for nodes in a stream).

Adding Models to Streams

To add the model nugget to your stream, click the icon in the model nuggets palette and then click the stream canvas where you want to place the node. Or right-click the icon and choose Add to Stream from the context menu. Then connect your stream to the node, and you are ready to pass data to generate predictions.

Caution: If you want to use a scoring nugget to regenerate a modeling node that contains both the category model and the template used, we recommend that you create a TAP and use it in an interactive session, in place of the modeling node, before generating the scoring nugget.

- [Category Model Nugget: Model Tab](#)
- [Category model nugget: Settings tab](#)
- [Category Model Nugget: Other Tabs](#)
- [Using category model nuggets in a stream](#)

Related information

- [Text Mining modeling node](#)
- [Mining for Concepts and Categories](#)
- [Text Mining Nugget: Concept Model](#)
- [Options for Including Concepts for Scoring](#)
- [Underlying Terms in Concept Models](#)
- [Text Mining model nugget: TMWBModelApplier](#)
- [Concept Model: Fields tab](#)
- [Concept Model: Summary Tab](#)
- [Category Model Nugget: Model Tab](#)
- [Category model nugget: Settings tab](#)
- [Category Model Nugget: Other Tabs](#)
- [Using category model nuggets in a stream](#)
- [Microsoft Internet Explorer settings for Help](#)

Category Model Nugget: Model Tab

For category models, the model tab displays the list of categories in the category model on the left and the descriptors for a selected category on the right. Each category is made up of a number of descriptors. For each category you select, the associated descriptors appear in the table. These descriptors can include concepts, category rules, types, and TLA patterns. The type of each descriptor, as well as some examples of what each descriptor represents, is also shown.

On this tab, the objective is to select the categories you want to use for scoring. For a category model, documents and records are scored into categories. If a document or record contains one or more of the descriptors in its text or any underlying terms, then that document or record is assigned to the category to which the descriptor belongs. These underlying terms include the synonyms defined in the linguistic resources (regardless of whether they were found in the text or not) as well as any extracted plural/singular terms found in the text used to generate the model nugget, permuted terms, terms from fuzzy grouping, and so on.

Note: If you generated a concept model nugget instead, this tab will contain different results. See the topic [Concept Model: Model Tab](#) for more information.

Category Tree

To learn more about each category, select that category and review the information that appears for the descriptors in that category. For each descriptor, you can review the following information:

- **Descriptor** name. This field contains an icon representing what kind of descriptor it is, as well as the descriptor name.

Table 1. Descriptor icons

		Concepts		TLA Patterns
		Types		Category Rules

- **Type.** This field contains the type name for the descriptor. Types are collections of similar concepts (semantic groupings), such as organization names, products, or positive opinions. Rules are not assigned to types.
- **Details.** This field contains a list of what is included in that descriptor. Depending on the number of matches, you may not see the entire list for each descriptor due to size limitations in the dialog box.

Selecting and Copying Categories

All top categories are selected for scoring by default, as shown in the check boxes in the left pane. A checked box means that the category will be used for scoring. An unchecked box means that the category will be excluded from scoring. You can check multiple rows by selecting them and clicking one of the check boxes in your selection. Also, if a category or subcategory is selected but one of its subcategories is not selected, then the checkbox shows a blue background to indicate that there is only a partial selection in the children of the selected category.

By right-clicking a category in the tree, you can display a context menu from which you can:

- **Check Selected.** Checks all check boxes for the selected rows in the table.
- **Uncheck Selected.** Unchecks all check boxes for the selected rows in the table.
- **Check All.** Checks all check boxes in the table. This results in all categories being used in the final output. You can also use the corresponding checkbox icon on the toolbar.
- **Uncheck All.** Unchecks all check boxes in the table. Unchecking a category means that it will not be used in the final output. You can also use the corresponding empty checkbox icon on the toolbar.

By right-clicking a cell in the descriptor table, you can display a context menu in which you can:

- **Copy.** The selected concept(s) are copied to the clipboard.
- **Copy With Fields.** The selected descriptor is copied to the clipboard along with the column headings.
- **Select All.** All rows in the table will be selected.

Related information

- [Concept Model: Fields tab](#)
 - [Concept Model: Summary Tab](#)
 - [Text Mining Nugget: Category Model](#)
 - [Category model nugget: Settings tab](#)
 - [Category Model Nugget: Other Tabs](#)
 - [Using category model nuggets in a stream](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Category model nugget: Settings tab

The Settings tab is used to define the text field value for the new input data, if necessary. It is also the place where you define the data model for your output (scoring mode).

Note: This tab appears in the node dialog box only when the model nugget is placed on the canvas or in a stream. It does not exist when you are accessing this nugget directly in the Models palette.

Scoring mode: Categories as fields

With this option, there are just as many output records as there were in the input. However, each record now contains one new field for every category that was selected (using the check mark) on the Model tab. For each field, enter a flag value for True and for False, such as Yes/No, True/False, T/F, or 1 and 2. The storage types are set automatically to reflect the values chosen. For example, if you enter numeric values for the flags, they will be automatically handled as an integer value. The storage types for flags can be string, integer, real number, or date/time.

Note: If you are using very large data sets, for example with a Db2 database, using Categories as fields may encounter processing problems due to the amount of data. In this case we recommend using Categories as records instead.

Field name extension. You can choose to specify an extension prefix/suffix for the field name or you can choose to use the category codes. Field names are generated by using the category name plus this extension.

- Add as. Specify where the extension should be added to the field name. Choose Prefix to add the extension to the beginning of the string. Choose Suffix to add the extension to the end of the string.

If a subcategory is unselected. This option allows you to specify how the descriptors belonging to subcategories that were not selected for scoring will be handled. There are two options.

- The option **Exclude its descriptors completely from scoring** will cause the descriptors of subcategories that do not have checkmarks (unselected) to be ignored and unused during scoring.
- The option **Aggregate descriptors with those in parent category** will cause the descriptors of subcategories that do not have checkmarks (unselected) to be used as descriptors for the parent category (the category above this subcategory). If several levels of subcategories are unselected, the descriptors will be rolled up under the first available parent category.

Score only lowest-level matching category. Use this option to output the category only on one single line (for example, if the category is **GeneralSatisfaction/Pos**, selecting this option results in **GeneralSatisfaction/Pos**. Without this option, you'll get two lines: **GeneralSatisfaction** and **GeneralSatisfaction/Pos**).

Accommodate punctuation errors. This option temporarily normalizes text containing punctuation errors (for example, improper usage) during extraction to improve the extractability of concepts. This option is extremely useful when text is short and of poor quality (as, for example, in open-ended survey responses, e-mail, and CRM data), or when the text contains many abbreviations.

Scoring mode: Categories as records

With this option, a new record is created for each **category**, **document** pair. Typically, there are more records in the output than there were in the input. In addition to the input fields, new fields are also added to the data depending on what kind of model it is.

Table 1. Output fields for "Categories as records"

New Output Field	Description
Category	Contains the category name to which the text document was assigned. If the categories is a subcategory of another, then the full path to the category name is controlled by the value you chose in this dialog.

Values for hierarchical categories. This option controls how the names of subcategories are displayed in the output.

- Full category path. This option will output the name of the category and the full path of parent categories if applicable using slashes to separate category names from subcategory names.
- Short category path. This option will output only the name of the category but use ellipses to show the number of parent categories for the category in question.
- Bottom level category. This option will output only the name of the category without the full path or parent categories shown.

If a **subcategory** is unselected. This option allows you to specify how the descriptors belonging to subcategories that were not selected for scoring will be handled. There are two options.

- The option **Exclude its descriptors completely from scoring** will cause the descriptors of subcategories that do not have checkmarks (unselected) to be ignored and unused during scoring.
- The option **Aggregate descriptors with those in parent category** will cause the descriptors of subcategories that do not have checkmarks (unselected) to be used as descriptors for the parent category (the category above this subcategory). If several levels of subcategories are unselected, the descriptors will be rolled up under the first available parent category.

Accommodate punctuation errors. This option temporarily normalizes text containing punctuation errors (for example, improper usage) during extraction to improve the extractability of concepts. This option is extremely useful when text is short and of poor quality (as, for example, in open-ended survey responses, e-mail, and CRM data), or when the text contains many abbreviations.

Category Model Nugget: Other Tabs

The Fields tab and Settings tab for the category model nugget are the same as for the concept model nugget.

- Fields tab. See the topic [Concept Model: Fields tab](#) for more information.
- Summary tab. See the topic [Concept Model: Summary Tab](#) for more information.

Related information

- [Concept Model: Fields tab](#)
- [Concept Model: Summary Tab](#)
- [Text Mining Nugget: Category Model](#)
- [Category Model Nugget: Model Tab](#)
- [Category model nugget: Settings tab](#)
- [Using category model nuggets in a stream](#)
- [Microsoft Internet Explorer settings for Help](#)

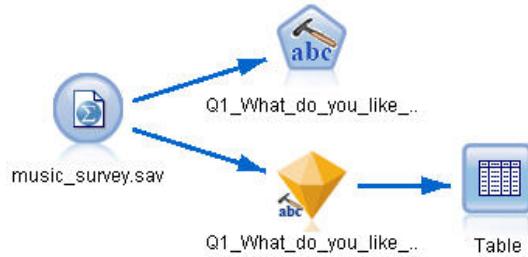
Using category model nuggets in a stream

The Text Mining category model nugget is generated from an interactive workbench session. You can use this model nugget in a stream.

Example: Statistics File node with the category model nugget

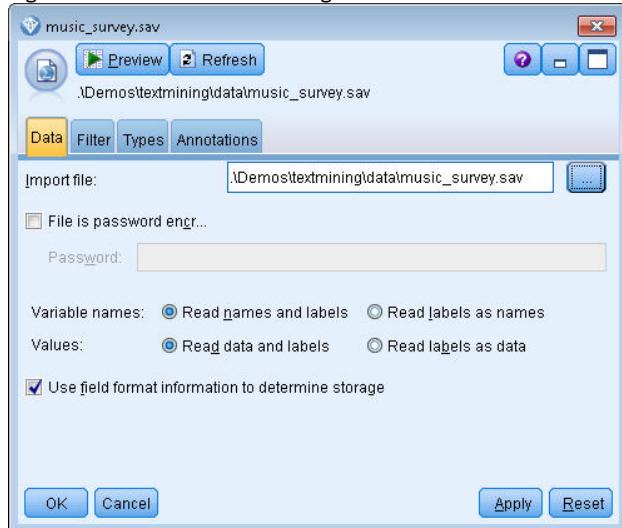
The following example shows how to use the Text Mining model nugget.

Figure 1. Example stream: Statistics File node with a Text Mining category model nugget



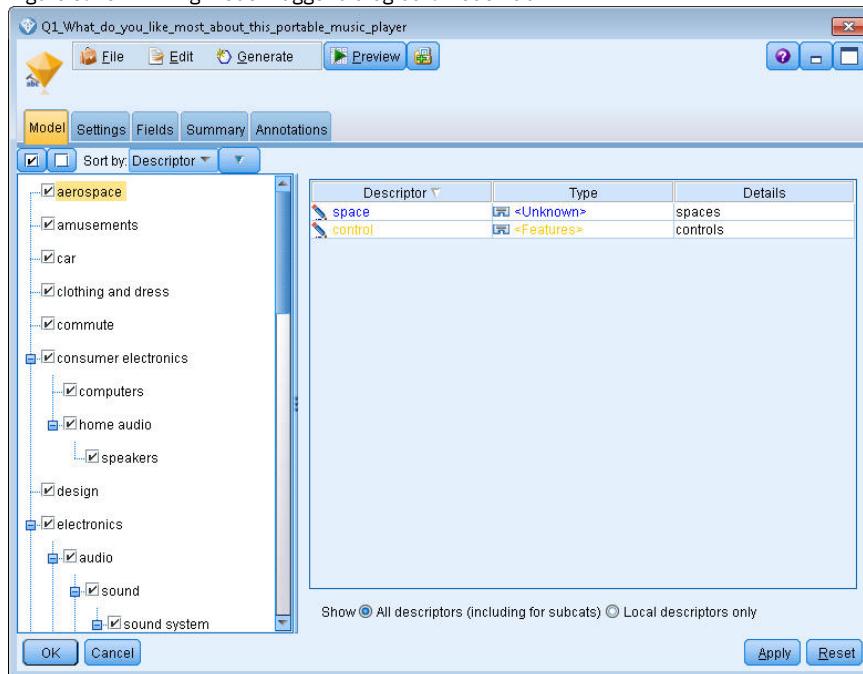
- 1. Statistics File node (Data tab).** First, we added this node to the stream to specify where the text documents are stored.

Figure 2. Statistics File node dialog box: Data tab



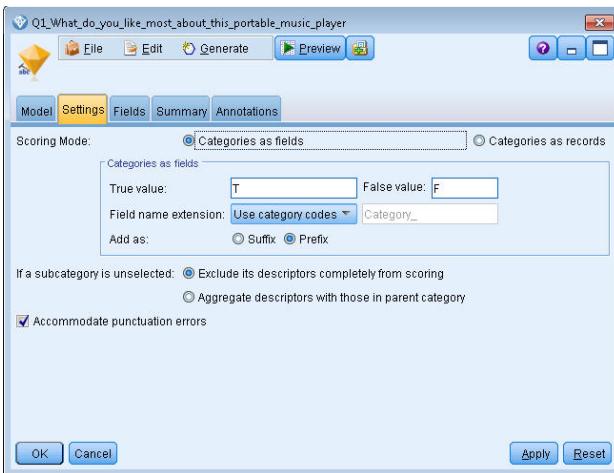
- 2. Text Mining category model nugget (Model tab).** Next, we added and connected a category model nugget to the Statistics File node. We selected the categories we wanted to use to score our data.

Figure 3. Text Mining model nugget dialog box: Model tab



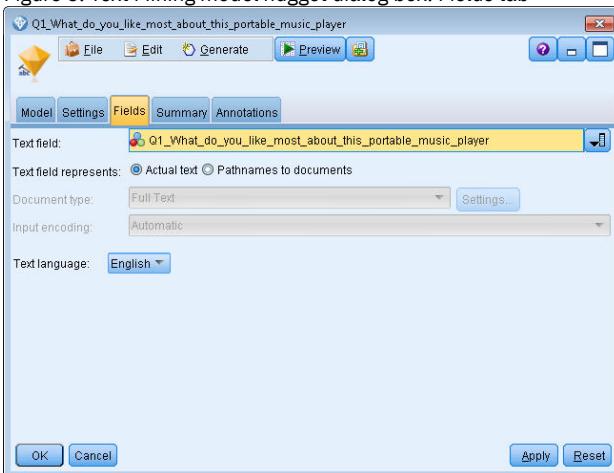
- 3. Text Mining model nugget (Settings tab).** Next, we defined the output format Categories as fields.

Figure 4. Category model nugget dialog box: Settings tab



4. **Text Mining category model nugget (Fields tab).** Next, we selected the text field variable, which is the field name coming from the Statistics File node, and selected the option Text field represents Actual text, as well as other settings.

Figure 5. Text Mining model nugget dialog box: Fields tab



5. **Table node.** Next, we attached a table node to see the results and executed the stream.

Figure 6. Table output

Table (3 fields, 844 records)			
	ID	Q1_What_do_you_like_most_about_this_portable_music_player	Category
1	1	little, light	light
2	2	The battery power is great.	light
3	2	The battery power is great.	electronics/battery
4	2	The battery power is great.	electronics
5	3	cost and size	size
6	6	Battery life. Portability. Accessories. Style.	light
7	6	Battery life. Portability. Accessories. Style.	electronics/battery
8	6	Battery life. Portability. Accessories. Style.	electronics
9	7	I like its ability to store all of my music. I also like the ability to create playlists.	playlists
10	7	I like its ability to store all of my music. I also like the ability to create playlists.	light
11	7	I like its ability to store all of my music. I also like the ability to create playlists.	music
12	8	portability, capacity, sound quality, durability	light
13	8	portability, capacity, sound quality, durability	electronics/audio/sound
14	8	portability, capacity, sound quality, durability	electronics/audio

Mining for Text Links

- [Text Link Analysis node](#)

Text Link Analysis node

The Text Link Analysis (TLA) node adds a pattern-matching technology to text mining's concept extraction in order to identify relationships between the concepts in the text data based on known patterns. These relationships can describe how a customer feels about a product, which companies are doing business together, or even the relationships between genes or pharmaceutical agents.

For example, extracting your competitor's product name may not be interesting enough to you. Using this node, you could also learn how people feel about this product, if such opinions exist in the data. The relationships and associations are identified and extracted by matching known patterns to your text data.

You can use the TLA pattern rules inside certain resource templates shipped with IBM® SPSS® Modeler Text Analytics or create/edit your own. Pattern rules are made up of macros, word lists, and word gaps to form a Boolean query, or rule, that is compared to your input text. Whenever a TLA pattern rule matches text, this text can be exacted as a TLA result and restructured as output data. See the topic [About Text Link Rules](#) for more information.

The Text Link Analysis node offers a more direct way to identify and extract TLA pattern results from your text and then add the results to the dataset in the stream. But the Text Link Analysis node is not the only way in which you can perform text link analysis. You can also use an interactive workbench session in the Text Mining modeling node.

In the interactive workbench, you can explore the TLA pattern results and use them as category descriptors and/or to learn more about the results using drill-down and graphs. See the topic [Exploring Text Link Analysis](#) for more information. In fact, using the Text Mining node to extract TLA results is a great way to explore and fine-tune templates to your data for later use directly in the TLA node.

The output can be represented in up to 6 slots, or parts. See the topic [TLA node output](#) for more information.

You can find this node on the IBM SPSS Modeler Text Analytics tab of nodes palette at the bottom of the IBM SPSS Modeler window. See the topic [IBM SPSS Modeler Text Analytics nodes](#) for more information.

Requirements. The Text Link Analysis node accepts text data read into a field using any of the standard source nodes (Database node, Flat File node, etc.) or read into a field listing paths to external documents generated by a File List node or a Web Feed node.

Strengths. The Text Link Analysis node goes beyond basic concept extraction to provide information about the relationships *between* concepts, as well as related opinions or qualifiers that may be revealed in the data.

- [Text Link Analysis node: Fields tab](#)
- [Text Link Analysis node: Expert tab](#)
- [TLA node output](#)
- [Caching TLA Results](#)
- [Using the Text Link Analysis node in a stream](#)

Text Link Analysis node: Fields tab

Use the Fields tab to specify the field settings for the data from which you will be extracting concepts. You can set the following parameters:

ID field. Select the field containing the identifier for the text records. Identifiers must be integers. The ID field serves as an index for the individual text records. Use an ID field if the text field represents the text to be mined.

Text field. Select the field containing the text to be mined. This field depends on the data source.

Language field. Select the field that contains the two letter ISO language identifier. If you do not select a field, the language of each document is assumed to be that of the supplied template.

Document type. The document type specifies the structure of the text. Select one of the following types:

- Full text. Use for most documents or text sources. The entire set of text is scanned for extraction. Unlike the other options, there are no additional settings for this option.
- Structured text. Use for bibliographic forms, patents, and any files that contain regular structures that can be identified and analyzed. This document type is used to skip all or part of the extraction process. It allows you to define term separators, assign types, and impose a minimum frequency value. If you select this option, you must click the Settings button and enter text separators in the Structured Text Formatting area of the Document Settings dialog box. See the topic [Document Settings for Fields Tab](#) for more information.

Textual unity. Select the extraction mode from the following:

- Document mode. Use for documents that are short and semantically homogenous, such as articles from news agencies.
- Paragraph mode. Use for Web pages and nontagged documents. The extraction process semantically divides the documents, taking advantage of characteristics such as internal tags and syntax. If this mode is selected, scoring is applied paragraph by paragraph.

Therefore, for example, the rule **apple & orange** is true only if **apple** and **orange** are found in the same paragraph.

Note: Due to the way text is extracted from PDF documents, Paragraph mode does not work on these documents. This is because the extraction suppresses the carriage return marker.

Paragraph mode settings. This option is available only if you set the textual unity option to Paragraph mode. Specify the character thresholds to be used in any extraction. The actual size is rounded up or down to the nearest period. To ensure that the word associations produced from the

text of the document collection are representative, avoid specifying an extraction size that is too small.

- Minimum. Specify the minimum number of characters to be used in any extraction.
- Maximum. Specify the maximum number of characters to be used in any extraction.

Copy resources from. When mining text, the extraction is based not only on the settings in the Expert tab but also on the linguistic resources. These resources serve as the basis for how to handle and process the text during extraction to get the concepts, types, and TLA patterns. You can copy resources into this node from a resource template.

A resource template is a predefined set of libraries and advanced linguistic and nonlinguistic resources that have been fine-tuned for a particular domain or usage. These resources serve as the basis for how to handle and process data during extraction. Click Load and selecting the template from which to copy your resources.

Templates are loaded when you select them and not when the stream is executed. At the moment that you load, a copy of the resources is stored into the node. Therefore, if you ever wanted to use an updated template, you would have to reload it here. See the topic [Copying resources from templates and TAPs](#) for more information.

Text language. Identifies the language of the text being mined. The resources copied in the node control the language options presented. Select the language for which the resources were tuned.

Text Link Analysis node: Expert tab

In this node, the extraction of text link analysis (TLA) pattern results is automatically enabled. The Expert tab contains certain additional parameters that impact how text is extracted and handled. The parameters in this dialog box control the basic behavior, as well as a few advanced behaviors, of the extraction process. There are also a number of linguistic resources and options that also impact the extraction results, which are controlled by the resource template you select.

Limit extraction to concepts with a global frequency of at least [n]. Specifies the minimum number of times a word or phrase must occur in the text in order for it to be extracted. In this way, a value of 5 limits the extraction to those words or phrases that occur at least five times in the entire set of records or documents.

In some cases, changing this limit can make a big difference in the resulting extraction results, and consequently, your categories. Let's say that you are working with some restaurant data and you do not increase the limit above 1 for this option. In this case, you might find *pizza* (1), *thin pizza* (2), *spinach pizza* (2), and *favorite pizza* (2) in your extraction results. However, if you were to limit the extraction to a global frequency of 5 or more and re-extract, you would no longer get three of these concepts. Instead you would get *pizza* (7), since *pizza* is the simplest form and also this word already existed as a possible candidate. And depending on the rest of your text, you might actually have a frequency of more than seven, depending on whether there are still other phrases with *pizza* in the text. Additionally, if *spinach pizza* was already a category descriptor, you might need to add *pizza* as a descriptor instead to capture all of the records. For this reason, change this limit with care whenever categories have already been created.

Note that this is an extraction-only feature; if your template contains terms (which they usually do), and a term for the template is found in the text, then the term will be indexed regardless of its frequency.

For example, suppose you use a Basic Resources template that includes "los angeles" under the <Location> type in the Core library; if your document contains Los Angeles only once, then Los Angeles will be part of the list of concepts. To prevent this you will need to set a filter to display concepts occurring at least the same number of times as the value entered in the Limit extraction to concepts with a global frequency of at least [n] field.

Accommodate punctuation errors. This option temporarily normalizes text containing punctuation errors (for example, improper usage) during extraction to improve the extractability of concepts. This option is extremely useful when text is short and of poor quality (as, for example, in open-ended survey responses, e-mail, and CRM data), or when the text contains many abbreviations.

Accommodate spelling for a minimum word character length of [n] This option applies a fuzzy grouping technique that helps group commonly misspelled words or closely spelled words under one concept. The fuzzy grouping algorithm temporarily strips all vowels (except the first one) and strips double/triple consonants from extracted words and then compares them to see if they are the same so that **modeling** and **modelling** would be grouped together. However, if each term is assigned to a different type, excluding the <Unknown> type, the fuzzy grouping technique will not be applied.

You can also define the minimum number of root characters required before fuzzy grouping is used. The number of root characters in a term is calculated by totaling all of the characters and subtracting any characters that form inflection suffixes and, in the case of compound-word terms, determiners and prepositions. For example, the term **exercises** would be counted as 8 root characters in the form "exercise," since the letter **s** at the end of the word is an inflection (plural form). Similarly, **apple sauce** counts as 10 root characters ("apple sauce") and **manufacturing of cars** counts as 16 root characters ("manufacturing car"). This method of counting is only used to check whether the fuzzy grouping should be applied but does not influence how the words are matched.

Note: If you find that certain words are later grouped incorrectly, you can exclude word pairs from this technique by explicitly declaring them in the Fuzzy Grouping: Exceptions section in the Advanced Resources tab. See the topic [Fuzzy Grouping](#) for more information.

Extract uniterms This option extracts single words (uniterms) as long as the word is not already part of a compound word and if it is either a noun or an unrecognized part of speech.

Extract nonlinguistic entities This option extracts nonlinguistic entities, such as phone numbers, social security numbers, times, dates, currencies, digits, percentages, e-mail addresses, and HTTP addresses. You can include or exclude certain types of nonlinguistic entities in the Nonlinguistic Entities: Configuration section of the Advanced Resources tab. By disabling any unnecessary entities, the extraction engine won't waste processing time. See the topic [Configuration](#) for more information.

Uppercase algorithm This option extracts simple and compound terms that are not in the built-in dictionaries as long as the first letter of the term is in uppercase. This option offers a good way to extract most proper nouns.

Group partial and full person names together when possible This option groups names that appear differently in the text together. This feature is helpful since names are often referred to in their full form at the beginning of the text and then only by a shorter version. This option attempts to match any uniterm with the <Unknown> type to the last word of any of the compound terms that is typed as <Person>. For example, if *doe* is found and initially typed as <Unknown>, the extraction engine checks to see if any compound terms in the <Person> type include *doe* as the last word, such as *john doe*. This option does not apply to first names since most are never extracted as uniterms.

Maximum nonfunction word permutation This option specifies the maximum number of nonfunction words that can be present when applying the permutation technique. This permutation technique groups similar phrases that differ from each other only by the nonfunction words (for example, *of* and *the*) contained, regardless of inflection. For example, let's say that you set this value to at most two words and both **company officials** and **officials of the company** were extracted. In this case, both extracted terms would be grouped together in the final concept list since both terms are deemed to be the same when *of the* is ignored.

Use derivation when grouping multiterms When processing Big Data, select this option to group multiterms by using derivation rules.

TLA node output

After running the Text Link Analysis node, the data are restructured. It is important to understand the way that text mining restructures your data. If you desire a different structure for data mining, you can use nodes on the Field Operations palette to accomplish this. For example, if you were working with data in which each row represented a text record, then one row is created for each pattern uncovered in the source text data. For each row in the output, there are 15 fields:

- Six fields (Concept#, such as Concept1, Concept2, ..., and Concept6) represent any concepts found in the pattern match.
- Six fields (Type#, such as Type1, Type2, ..., and Type6) represent the type for each concept.
- Rule Name represents the name of the text link rule used to match the text and produce the output.
- A field using the name of the ID field you specified in the node and representing the record or document ID as it was in the input data
- Matched Text represents the portion of the text data in the original record or document that was matched to the TLA pattern.

Note: Any preexisting streams containing a Text Link Analysis node from a release prior to 5.0 may not be fully executable until you update the nodes. Certain improvements in later versions of IBM® SPSS® Modeler require older nodes to be replaced with the newer versions, which are both more deployable and more powerful.

It is also possible to perform an automatic translation of certain languages. This feature enables you to mine documents in a language you may not speak or read. If you want to use the translation feature, you must have access to the SDL Software as a Service (SaaS). See the topic [Translation Settings](#) for more information.

Caching TLA Results

If you cache, the text link analysis results are in the stream. To avoid repeating the extraction of text link analysis results each time the stream is executed, select the Text Link Analysis node and from the menus choose, Edit > Node > Cache > Enable. The next time the stream is executed, the output is cached in the node. The node icon displays a tiny "document" graphic that changes from white to green when the cache is filled. The cache is preserved for the duration of the session. To preserve the cache for another day (after the stream is closed and reopened), select the node and from the menus choose, Edit > Node > Cache > Save Cache. The next time you open the stream, you can reload the saved cache rather than running the translation again.

Alternatively, you can save or enable a node cache by right-clicking the node and choosing Cache from the context menu.

Using the Text Link Analysis node in a stream

The Text Link Analysis node is used to access data and extract concepts in a stream. You can use any source node to access data.

Example: Statistics File node with the Text Link Analysis node

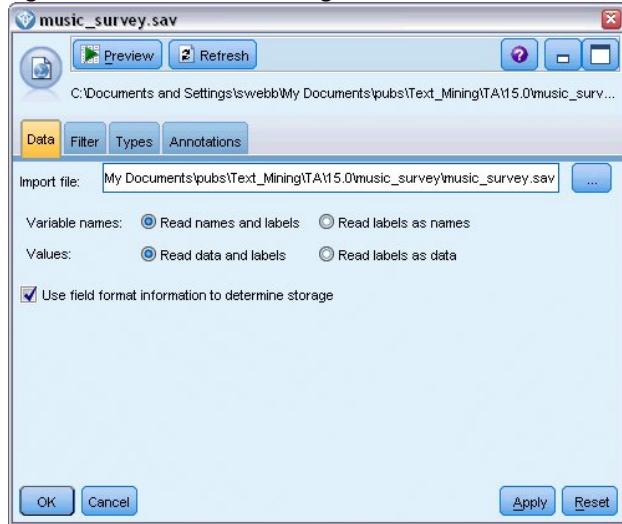
The following example shows how to use the Text Link Analysis node.

Figure 1. Example: Statistics File node with the Text Link Analysis node



1. Statistics File node (Data tab). First, we added this node to the stream to specify where the text is stored.

Figure 2. Statistics File node dialog box: Data tab



2. Text Link Analysis node (Fields tab). Next, we attached this node to the stream to extract concepts for downstream modeling or viewing. We specified the ID field and the text field name containing the data, as well as other settings.

Figure 3. Text Link Analysis node dialog box: Fields tab



3. Table node. Finally, we attached a Table node to view the concepts that were extracted from our text documents. In the table output shown, you can see the TLA pattern results found in the data after this stream was executed with a Text Link Analysis node. Some results show only one concept/type was matched. In others, the results are more complex and contain several types and concepts. Additionally, as a result of running data through the Text Link Analysis node and extracting concepts, several aspects of the data are changed. The original data in our example contained 8 fields and 405 records. After executing the Text Link Analysis node, there are now 15 fields and 640 records. There is now one row for each TLA pattern result found. For example, **ID 7** became three rows from the original because three TLA pattern results were extracted. You can use a Merge node if you want to merge this output data back into your original data.

Figure 4. Table output node

The screenshot shows a 'Table (15 fields, 640 records) #4' window. The 'Annotations' tab is selected. The preview pane displays 10 rows of data with their respective annotations:

	Concept1	Type1	Concept2	Type2	Concept3	Type3	Concept4	Type4	Concept5	Type5	Concept6	Type6	Rule Number	ID	Matched Text
1	expensive	Negative	Budget	Null	Null	Null	Null	Null	Null	Null	Null	001350	opinion	1	<expensive>
2	screen	Unknown	difficult	Nega...	Null	Null	Null	Null	Null	Null	Null	00145	topic + opinion	2	The <screen> is <hard> to see when outside
3	software	Unknown	difficult	Nega...	Null	Null	Null	Null	Null	Null	Null	00211	_opinion + topic	3	<difficult><software>
4	nothing	Uncertain	Null	Null	Null	Null	Null	Null	Null	Null	Null	00153	_topic+opinion	4	<Nothing> <*> I love it
5	like	Positive	Null	Null	Null	Null	Null	Null	Null	Null	Null	00350	_opinion	4	Nothing , <I love it>
6	battery life	Unknown	too long	Nega...	Null	Null	Null	Null	Null	Null	Null	00145	_topic + opinion	5	<Battery life> seems <shorter> than advertised
7	ubiquitousness	Unknown	Null	Null	Null	Null	Null	Null	Null	Null	Null	00500	_topic	6	<Ubiquitousness>
8	40gb model	Unknown	available	Posit...	Null	Null	Null	Null	Null	Null	Null	00145	_topic + opinion	7	I wish the <40GB model> was still <available>
9	20gb model	Unknown	Null	Null	Null	Null	Null	Null	Null	Null	Null	00102	_topic + Negative + topic	7	I have a <200B model> and <need more> <memory>
10	memory	Unknown	need more	Nega...	Null	Null	Null	Null	Null	Null	Null	00102	_topic + Negative + topic	7	I have a <200B model> and <need more> <memory>

Browsing External Source Text

- [File Viewer node](#)

File Viewer node

When you are mining a collection of documents, you can specify the full path names of files directly into your Text Mining modeling nodes. However, when outputting to a Table node, you will only see the full path name of a document rather than the text within it. The File Viewer node can be used as an analog of the Table node and it enables you to access the actual text within each of the documents without having to merge them all together into a single file.

The File Viewer node can help you better understand the results from text extraction by providing you access to the source, or untranslated, text from which concepts were extracted since it is otherwise inaccessible in the stream. This node is added to the stream after a File List node to obtain a list of links to all the files.

The result of this node is a window showing all of the document elements that were read and used to extract concepts. From this window, you can click a toolbar icon to launch the report in an external browser listing document names as hyperlinks. You can click a link to open the corresponding document in the collection. See the topic [Using the File Viewer Node](#) for more information.

You can find this node on the IBM® SPSS® Modeler Text Analytics tab of nodes palette at the bottom of the IBM SPSS Modeler window. See the topic [IBM SPSS Modeler Text Analytics nodes](#) for more information.

Note: When you are working in client-server mode and File Viewer nodes are part of the stream, document collections must be stored in a Web server directory on the server. Since the Text Mining output node produces a list of documents stored in the Web server directory, the Web server's security settings manage the permissions to these documents.

- [File Viewer Node Settings](#)
- [Using the File Viewer Node](#)

File Viewer Node Settings

You can specify the following settings for the File Viewer node.

Document field. Select the field from your data that contains the full name and path of the documents to be displayed.

Title for generated HTML page. Create a title to appear at the top of the page that contains the list of documents.

Related information

- [File Viewer node](#)
- [Using the File Viewer Node](#)
- [Microsoft Internet Explorer settings for Help](#)

Using the File Viewer Node

The following example shows how to use the File Viewer node.

Example: File List node and a File Viewer node

Figure 1. Stream illustrating the use of a File Viewer node



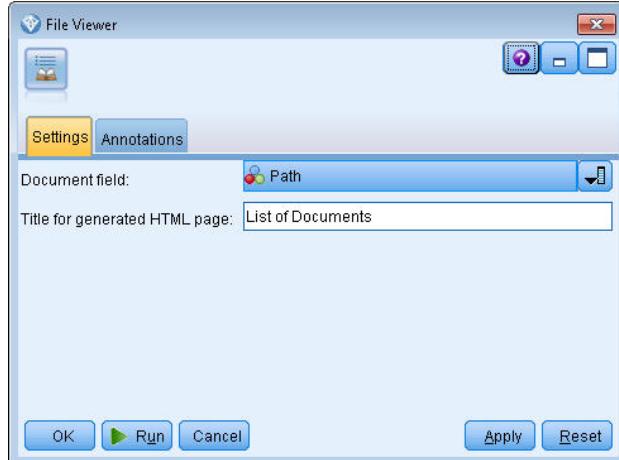
1. **File List node (Settings tab).** First, we added this node to specify where the documents are located.

Figure 2. File List node dialog box: Settings tab



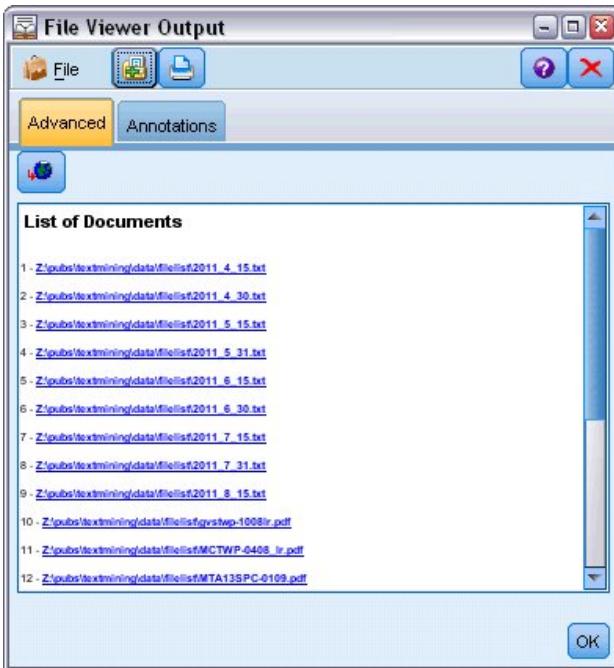
2. **File Viewer node (Settings tab).** Next, we attached the File Viewer node to produce an HTML list of documents.

Figure 3. File Viewer node dialog box: Settings tab



3. **File Viewer Output dialog.** Next, we executed the stream which outputs the list of documents in a new window.

Figure 4. File Viewer Output



4. To see the documents, we clicked the toolbar button showing a globe with a red arrow. This opened a list of document hyperlinks in our browser.

Related information

- [File Viewer node](#)
- [File Viewer Node Settings](#)
- [Microsoft Internet Explorer settings for Help](#)

Node Properties for Scripting

IBM® SPSS® Modeler has a scripting language to allow you to execute streams from the command line. Here, you can learn about the node properties that are specific to each of the nodes delivered with IBM SPSS Modeler Text Analytics. For more information on the standard set of nodes delivered with IBM SPSS Modeler, please refer to the Scripting and Automation Guide.

- [File List Node: filelistnode](#)
- [Web Feed Node: webfeednode](#)
- [Language Node: languageidentifier](#)
- [Text Mining node: TextMiningWorkbench](#)
- [Text Mining model nugget: TMWBModelApplier](#)
- [Text Link Analysis node: textlinkanalysis](#)

Related information

- [Mining for Concepts and Categories](#)
- [Text Mining modeling node](#)
- [Text Mining Node: Fields Tab](#)
- [Document Settings for Fields Tab](#)
- [Text Mining Node: Model Tab](#)
- [Build Interactively](#)
- [Generate Directly](#)
- [Copying resources from templates and TAPs](#)
- [Text Mining node: Expert tab](#)
- [Sampling Upstream to Save Time](#)
- [Using the Text Mining node in a stream](#)
- [Text Mining node: TextMiningWorkbench](#)
- [Microsoft Internet Explorer settings for Help](#)

File List Node: filelistnode

You can use the properties in the following table for scripting. The node itself is called `filelistnode`.

Table 1. File List node scripting properties

Scripting properties	Data type
<code>path</code>	<code>string</code>
<code>recurse</code>	<code>flag</code>
<code>word_processing</code>	<code>flag</code>
<code>excel_file</code>	<code>flag</code>
<code>powerpoint_file</code>	<code>flag</code>
<code>text_file</code>	<code>flag</code>
<code>web_page</code>	<code>flag</code>
<code>xml_file</code>	<code>flag</code>
<code>pdf_file</code>	<code>flag</code>
<code>no_extension</code>	<code>flag</code>

Note: 'Create list' parameter is no longer available and any scripts containing that option will be automatically converted into a 'Files' output.

Related information

- [File List node](#)
- [File List node: Settings tab](#)
- [File List Node: Other Tabs](#)
- [Using the File List node in text mining](#)
- [Microsoft Internet Explorer settings for Help](#)

Web Feed Node: webfeednode

You can use the properties in the following table for scripting. The node itself is called `webfeednode`.

Table 1. Web Feed node scripting properties

Scripting properties	Data type	Property description
<code>urls</code>	<code>string1 string2 ...stringn</code>	Each URL is specified in the list structure. URL list separated by “\n”
<code>recent_entries</code>	<code>flag</code>	
<code>limit_entries</code>	<code>integer</code>	Number of most recent entries to read per URL.
<code>use_previous</code>	<code>flag</code>	To save and reuse Web feed cache.
<code>use_previous_label</code>	<code>string</code>	Name for the saved Web cache.
<code>start_record</code>	<code>string</code>	Non-RSS start tag.
<code>url n .title</code>	<code>string</code>	For each URL in the list, you must define one here too. The first one will be <code>url1.title</code> , where the number matches its position in the URL list. This is the start tag containing the title of the content.
<code>url n .short_description</code>	<code>string</code>	Same as for <code>url n .title</code> .
<code>url n .description</code>	<code>string</code>	Same as for <code>url n .title</code> .
<code>url n .authors</code>	<code>string</code>	Same as for <code>url n .title</code> .
<code>url n .contributors</code>	<code>string</code>	Same as for <code>url n .title</code> .
<code>url n .published_date</code>	<code>string</code>	Same as for <code>url n .title</code> .

Scripting properties	Data type	Property description
<code>url_n.modified_date</code>	<code>string</code>	Same as for <code>url_n.title</code> .
<code>html_alg</code>	<code>None HTMLCleaner</code>	Content filtering method.
<code>discard_lines</code>	<code>flag</code>	Discard short lines. Used with <code>min_words</code>
<code>min_words</code>	<code>integer</code>	Minimum number of words.
<code>discard_words</code>	<code>flag</code>	Discard short lines. Used with <code>min_avg_len</code>
<code>min_avg_len</code>	<code>integer</code>	
<code>discard_scw</code>	<code>flag</code>	Discard lines with many single character words. Used with <code>max_scw</code>
<code>max_scw</code>	<code>integer</code>	Maximum proportion 0-100 percentage of single characters words in a line
<code>discard_tags</code>	<code>flag</code>	Discard lines containing certain tags.
<code>tags</code>	<code>string</code>	Special characters must be escaped with a backslash character \.
<code>discard_spec_words</code>	<code>flag</code>	Discard lines containing specific strings
<code>words</code>	<code>string</code>	Special characters must be escaped with a backslash character \.

Related information

- [Web Feed node](#)
- [Web Feed Node: Input Tab](#)
- [Web Feed Node: Records Tab](#)
- [Web Feed Node: Content Filter Tab](#)
- [Using the Web Feed Node in Text Mining](#)
- [Microsoft Internet Explorer settings for Help](#)

Language Node: languageidentifier

You can use the properties in the following table for scripting. The node itself is called `languageidentifier`.

Table 1. Language node scripting properties

Scripting properties	Data type	Property description
<code>text</code>	<code>field</code>	
<code>language_field_name</code>	<code>string</code>	The field name that is generated as output.
<code>unidentified_language_value</code>	<code>Undefined Supported Custom</code>	Default value to be used when the language cannot be identified.
<code>unidentified_language_supported</code>	<code>en de es fr it ja nl pt</code>	Iso code. Only available if <code>unidentified_language_value</code> is <code>Supported</code> .
<code>unidentified_language_custom</code>	<code>string</code>	Only available if <code>unidentified_language_value</code> is <code>Custom</code> .

Related information

- [Language Node](#)
- [Language Node: Settings Tab](#)
- [Microsoft Internet Explorer settings for Help](#)

Text Mining node: TextMiningWorkbench

You can use the following parameters to define or update a node through scripting. The node itself is called `TextMiningWorkbench`.

Important: It is not possible to specify a different resource template via scripting. If you think you need a template, you must select it in the node dialog box.

Table 1. Text Mining modeling node scripting properties

Scripting properties	Data type	Property description
<code>text</code>	<code>field</code>	

Scripting properties	Data type	Property description
<code>method</code>	<code>ReadText ReadPath</code>	
<code>docType</code>	<code>integer</code>	With possible values (0,1,2) where 0 = <code>Full Text</code> , 1 = <code>Structured Text</code> , and 2 = <code>XML</code>
<code>encoding</code>	<code>Automatic "UTF-8" "UTF-16" "ISO-8859-1" "US-ASCII" "CP850" "EUC-JP" "SHIFT-JIS" "ISO2022-JP"</code>	Note that values with special characters, such as " <code>UTF-8</code> ", should be quoted to avoid confusion with a mathematical operator.
<code>unity</code>	<code>integer</code>	With possible values (0,1) where 0 = <code>Paragraph</code> and 1 = <code>Document</code>
<code>para_min</code>	<code>integer</code>	
<code>para_max</code>	<code>integer</code>	
<code>mtag</code>	<code>string</code>	Contains all the mtag settings (from Settings dialog box for XML files)
<code>mclef</code>	<code>string</code>	Contains all the mclef settings (from Settings dialog box for Structured Text files)
<code>partition</code>	<code>field</code>	
<code>custom_field</code>	<code>flag</code>	Indicates whether or not a partition field will be specified.
<code>use_model_name</code>	<code>flag</code>	
<code>model_name</code>	<code>string</code>	
<code>use_partitioned_data</code>	<code>flag</code>	If a partition field is defined, only the training data are used for model building.
<code>model_output_type</code>	<code>Interactive Model</code>	<code>Interactive</code> results in a category model. <code>Model</code> results in a concept model.
<code>use_interactive_info</code>	<code>flag</code>	For building interactively in a workbench session only.
<code>reuse_extraction_results</code>	<code>flag</code>	For building interactively in a workbench session only.
<code>interactive_view</code>	<code>Categories TIA Clusters</code>	For building interactively in a workbench session only.
<code>extract_top</code>	<code>integer</code>	This parameter is used when <code>model_type = Concept</code>
<code>use_check_top</code>	<code>flag</code>	
<code>check_top</code>	<code>integer</code>	
<code>use_uncheck_top</code>	<code>flag</code>	
<code>uncheck_top</code>	<code>integer</code>	
<code>language</code>	<code>de en es fr it ja nl pt</code>	
<code>frequency_limit</code>	<code>integer</code>	Deprecated in 14.0.
<code>concept_count_limit</code>	<code>integer</code>	Limit extraction to concepts with a global frequency of at least this value.
<code>fix_punctuation</code>	<code>flag</code>	
<code>fix_spelling</code>	<code>flag</code>	
<code>spelling_limit</code>	<code>integer</code>	
<code>extract_uniterm</code>	<code>flag</code>	
<code>extract_nonlinguistic</code>	<code>flag</code>	
<code>upper_case</code>	<code>flag</code>	
<code>group_names</code>	<code>flag</code>	
<code>permutation</code>	<code>integer</code>	Maximum nonfunction word permutation (the default is 3).

Text Mining model nugget: TMWBModelApplier

You can use the properties in the following table for scripting. The nugget itself is called `TMWBModelApplier`.

Table 1. Text Mining Model Nugget Properties

Scripting properties	Data type	Property description
----------------------	-----------	----------------------

Scripting properties	Data type	Property description
<code>scoring_mode</code>	<code>Fields Records</code>	
<code>field_values</code>	<code>Flags Counts</code>	This option is not available in the Category model nugget. For <code>Flags</code> , set to <code>TRUE</code> or <code>FALSE</code>
<code>true_value</code>	<code>string</code>	With <code>Flags</code> , define the value for true.
<code>false_value</code>	<code>string</code>	With <code>Flags</code> , define the value for false.
<code>extension_concept</code>	<code>string</code>	Specify an extension for the field name. Field names are generated by using the concept name plus this extension. Specify where to put this extension using the <code>add_as</code> value.
<code>extension_category</code>	<code>string</code>	Field name extension. You can choose to specify an extension prefix/suffix for the field name or you can choose to use the category codes. Field names are generated by using the category name plus this extension. Specify where to put this extension using the <code>add_as</code> value.
<code>add_as</code>	<code>Suffix Prefix</code>	
<code>fix_punctuation</code>	<code>flag</code>	
<code>excluded_subcategories_descriptors</code>	<code>RollUpToParent Ignore</code>	For category models only. If a subcategory is unselected. This option allows you to specify how the descriptors belonging to subcategories that were not selected for scoring will be handled. There are two options. <ul style="list-style-type: none"> Ignore. The option Exclude its descriptors completely from scoring will cause the descriptors of subcategories that do not have checkmarks (unselected) to be ignored and unused during scoring. RollUpToParent. The option Aggregate descriptors with those in parent category will cause the descriptors of subcategories that do not have checkmarks (unselected) to be used as descriptors for the parent category (the category above this subcategory). If several levels of subcategories and unselected, the descriptors will be rolled up under the first available parent category
<code>check_model</code>	<code>flag</code>	Deprecated in version 14
<code>text</code>	<code>field</code>	
<code>method</code>	<code>ReadText ReadPath</code>	
<code>docType</code>	<code>integer</code>	With possible values (0,1,2) where 0 = <code>Full Text</code> , 1 = <code>Structured Text</code> , and 2 = <code>XML</code>
<code>encoding</code>	<code>Automatic "UTF-8" "UTF-16" "ISO-8859-1" "US-ASCII" "CP850" "EUC-JP" "SHIFT-JIS" "ISO2022-JP"</code>	Note that values with special characters, such as " <code>UTF-8</code> ", should be quoted to avoid confusion with a mathematical operator.
<code>language</code>	<code>de en es fr it ja nl pt</code>	

Text Link Analysis node: `textlinkanalysis`

You can use the parameters in the following table to define or update a node through scripting. The node itself is called `textlinkanalysis`.

Important: It is not possible to specify a resource template via scripting. To select a template, you must do so from within the node dialog box.

Table 1. Text Link Analysis (TLA) node scripting properties

Scripting properties	Data type	Property description
<code>id_field</code>	<code>field</code>	
<code>text</code>	<code>field</code>	
<code>method</code>	<code>ReadText ReadPath</code>	
<code>docType</code>	<code>integer</code>	With possible values (0,1,2) where 0 = <code>Full Text</code> , 1 = <code>Structured Text</code> , and 2 = <code>XML</code>
<code>encoding</code>	<code>Automatic "UTF-8" "UTF-16" "ISO-8859-1" "US-ASCII" "CP850" "EUC-JP" "SHIFT-JIS" "ISO2022-JP"</code>	Note that values with special characters, such as " <code>UTF-8</code> ", should be quoted to avoid confusion with a mathematical operator.
<code>unity</code>	<code>integer</code>	With possible values (0,1) where 0 = <code>Paragraph</code> and 1 = <code>Document</code>
<code>para_min</code>	<code>integer</code>	
<code>para_max</code>	<code>integer</code>	
<code>mtag</code>	<code>string</code>	Contains all the mtag settings (from Settings dialog box for XML files)

Scripting properties	Data type	Property description
<code>mclef</code>	<code>string</code>	Contains all the mclef settings (from Settings dialog box for Structured Text files)
<code>language</code>	<code>de en es fr it ja nl pt</code>	
<code>concept_count_limit</code>	<code>integer</code>	Limit extraction to concepts with a global frequency of at least this value.
<code>fix_punctuation</code>	<code>flag</code>	
<code>fix_spelling</code>	<code>flag</code>	
<code>spelling_limit</code>	<code>integer</code>	
<code>extract_uniterm</code>	<code>flag</code>	
<code>extract_nonlinguistic</code>	<code>flag</code>	
<code>upper_case</code>	<code>flag</code>	
<code>group_names</code>	<code>flag</code>	
<code>permutation</code>	<code>integer</code>	Maximum nonfunction word permutation (the default is 3).

Interactive workbench mode

From a text mining modeling node, you can choose to launch an interactive workbench session during stream execution. In this workbench, you can extract key concepts from your text data, build categories, and explore text link analysis patterns and clusters, and generate category models. In this section, we discuss the workbench interface from a high-level perspective along with the major elements with which you will work, including:

- Extraction results. After an extraction is performed, these are the key words and phrases identified and extracted from your text data, also referred to as *concepts*. These concepts are grouped into *types*. Using these concepts and types, you can explore your data as well as create your categories. These are managed in the Categories and Concepts view.
- Categories. Using descriptors (such as extraction results, patterns, and rules) as a definition, you can manually or automatically create a set of categories to which documents and records are assigned based on whether or not they contain a part of the category definition. These are managed in the Categories and Concepts view.
- Clusters. *Clusters* are a grouping of concepts between which links have been discovered that indicate a relationship among them. The concepts are grouped using a complex algorithm that uses, among other factors, how often two concepts appear together compared to how often they appear separately. These are managed in the Clusters view. You can also add the concepts that make up a cluster to categories.
- Text link analysis patterns. If you have text link analysis (TLA) pattern rules in your linguistic resources or are using a resource template that already has some TLA rules, you can extract patterns from your text data. These patterns can help you uncover interesting relationships between concepts in your data. You can also use these patterns as descriptors in your categories. These are managed in the **Text Link Analysis** view.
- Linguistic resources. The extraction process relies on a set of parameters and linguistic definitions to govern how text is extracted and handled. These are managed in the form of templates and libraries in the Resource Editor view.

Potential Interactive Workbench issues

- Multiple Interactive Workbench sessions can cause sluggish behavior. SPSS® Modeler Text Analytics and SPSS Modeler share a common Java run-time engine when an interactive workbench session is launched. Depending on the number of Interactive Workbench sessions you invoke during a SPSS Modeler session, system memory may cause the application to become sluggish, even if opening and closing the same session. This effect may be especially pronounced if you are working with large data or have a machine with less than the recommended RAM setting of 4GB. If you notice your machine is slow to respond, it is recommended that you save all your work, shut down SPSS Modeler, and re-launch the application. Running SPSS Modeler Text Analytics on a machine with less than the recommended memory, particularly when working with large data sets or for prolonged periods of time, may cause Java to run out of memory and shut down. It is strongly suggested you upgrade to the recommended memory setting or larger (or use SPSS Modeler Text Analytics Server) if you work with large data.
- SPSS Modeler Client can run out of memory after multiple SPSS Modeler Text Analytics Interactive Workbench sessions are run without restarting the application. Monitor the memory usage in the status line and, if running low, close and re-open SPSS Modeler Client.

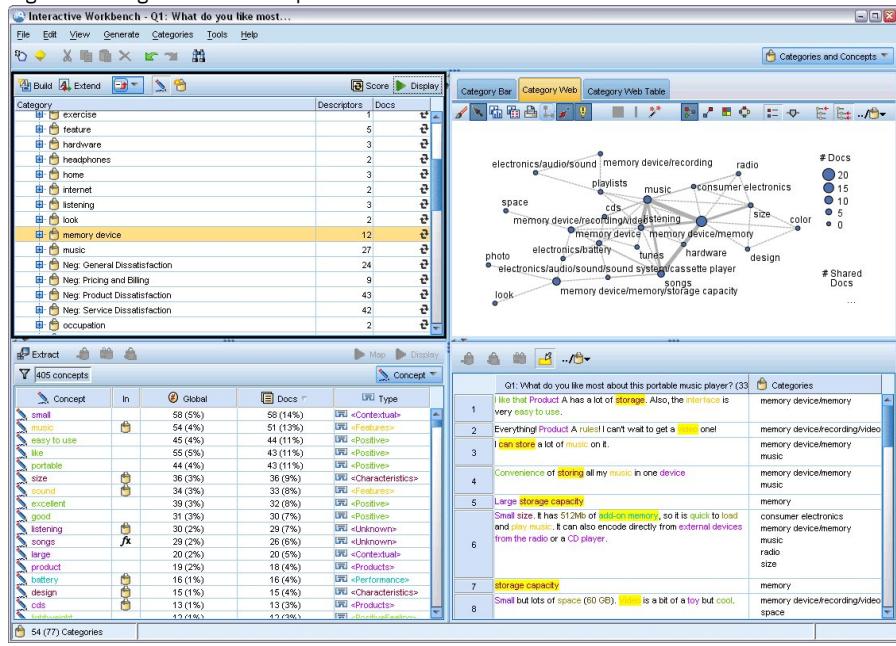
- [The Categories and Concepts View](#)
- [The Clusters View](#)
- [The Text Link Analysis view](#)
- [The Resource Editor view](#)
- [Setting Options](#)
- [Microsoft Internet Explorer settings for Help](#)
- [Generating Model Nuggets and Modeling Nodes](#)

- [Updating Modeling Nodes and Saving](#)
- [Closing and Ending Sessions](#)
- [Keyboard Accessibility](#)

The Categories and Concepts View

The application interface is made up of several views. The Categories and Concepts view is the window in which you can create and explore categories as well as explore and tweak the extraction results. *Categories* refers to a group of closely related ideas and patterns to which documents and records are assigned through a scoring process. While *concepts* refer to the most basic level of extraction results available to use as building blocks, called descriptors, for your categories.

Figure 1. Categories and Concepts view



The Categories and Concepts view is organized into four panes, each of which can be hidden or shown by selecting its name from the View menu. See the topic [Categorizing text data](#) for more information.

Categories Pane

Located in the upper left corner, this area presents a table in which you can manage any categories you build. After extracting the concepts and types from your text data, you can begin building categories by using techniques such as semantic networks and concept inclusion, or by creating them manually. If you double-click a category name, the Category Definitions dialog box opens and displays all of the descriptors that make up its definition, such as concepts, types, and rules. See the topic [Categorizing text data](#) for more information. Not all automatic techniques are available for all languages.

When you select a row in the pane, you can then display information about corresponding documents/records or descriptors in the Data and Visualization panes.

Extraction Results Pane

Located in the lower left corner, this area presents the extraction results. When you run an extraction, the extraction engine reads through the text data, identifies the relevant concepts, and assigns a type to each. *Concepts* are words or phrases extracted from your text data. *Types* are semantic groupings of concepts stored in the form of type dictionaries. When the extraction is complete, concepts and types appear with color coding in the Extraction Results pane. See the topic [Extraction results: Concepts and types](#) for more information.

You can see the set of underlying terms for a concept by hovering your mouse over the concept name. Doing so will display a tooltip showing the concept name and up to several lines of terms that are grouped under that concept. These underlying terms include the synonyms defined in the linguistic resources (regardless of whether they were found in the text or not) as well as the any extracted plural/singular terms, permuted terms, terms from fuzzy grouping, and so on. You can copy these terms or see the full set of underlying terms by right-clicking the concept name and choosing the context menu option.

Text mining is an iterative process in which extraction results are reviewed according to the context of the text data, fine-tuned to produce new results, and then reevaluated. Extraction results can be refined by modifying the linguistic resources. This fine-tuning can be done in part directly

from the Extraction Results or Data pane but also directly in the Resource Editor view. See the topic [The Resource Editor view](#) for more information.

Note: If there are more results that can fit in the visible pane, you can use the controls at the bottom of the pane to move forwards and backwards through the results, or enter a page number to go to.

Visualization Pane

Located in the upper right corner, this area presents multiple perspectives on the commonalities in document/record categorization. Each graph or chart provides similar information but presents it in a different manner or with a different level of detail. These charts and graphs can be used to analyze your categorization results and aid in fine-tuning categories or reporting. For example, in a graph you might uncover categories that are too similar (for example, they share more than 75% of their records) or too distinct. The contents in a graph or chart correspond to the selection in the other panes. See the topic [Category Graphs and Charts](#) for more information.

Data Pane

The Data pane is located in the lower right corner. This pane presents a table containing the documents or records corresponding to a selection in another area of the view. Depending on what is selected, only the corresponding text appears in the Data pane. Once you make a selection, click a Display button to populate the Data pane with the corresponding text.

If you have a selection in another pane, the corresponding documents or records show the concepts highlighted in color to help you easily identify them in the text. You can also hover your mouse over color-coded items to display a tooltip showing name of the concept under which it was extracted and the type to which it was assigned. See the topic [The Data Pane](#) for more information.

Searching and Finding in the Categories and Concepts view

In some cases, you may need to locate information quickly in a particular section. Using the Find toolbar, you can enter the string you want to search for and define other search criteria such as case sensitivity or search direction. Then you can choose the pane in which you want to search.

To use the Find feature

1. In the Categories and Concepts view, choose Edit > Find from the menus. The Find toolbar appears above the Categories pane and Visualization panes.
2. Enter the word string that you want to search for in the text box. You can use the toolbar buttons to control the case sensitivity, partial matching, and direction of the search.
3. In the toolbar, click the name of the pane in which you want to search. If a match is found, the text is highlighted in the window.
4. To look for the next match, click the name of the pane again.

The Clusters View

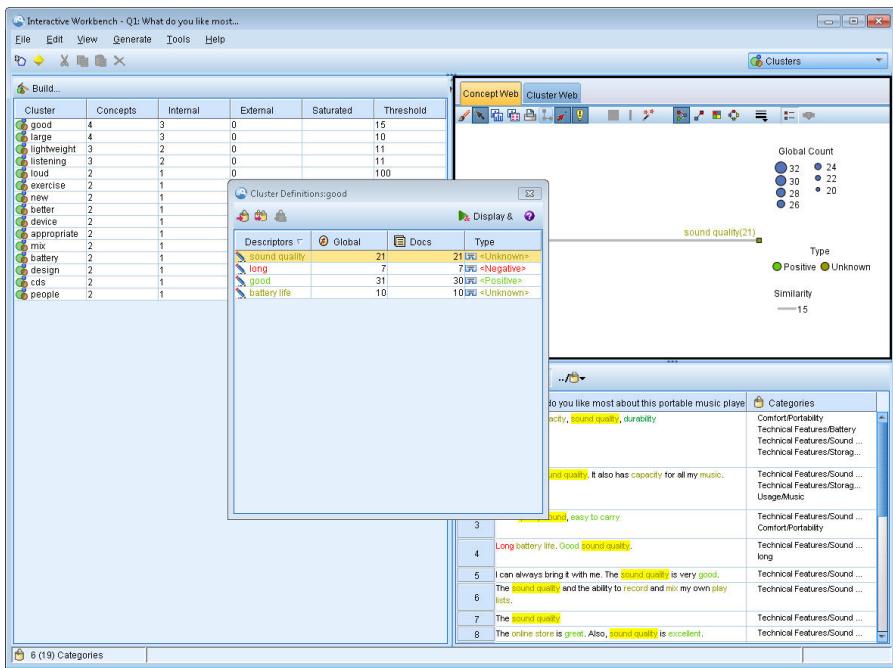
In the Clusters view, you can build and explore cluster results found in your text data. *Clusters* are groupings of concepts generated by clustering algorithms based on how often concepts occur and how often they appear together. The goal of clusters is to group concepts that co-occur together while the goal of categories is to group documents or records based on how the text they contain matches the descriptors (concepts, rules, patterns) for each category.

The more often the concepts within a cluster occur together coupled with the less frequently they occur with other concepts, the better the cluster is at identifying interesting concept relationships. Two concepts co-occur when they both appear (or one of their synonyms or terms appear) in the same document or record. See the topic [Analyzing clusters](#) for more information.

You can build clusters and explore them in a set of charts and graphs that could help you uncover relationships among concepts that would otherwise be too time-consuming to find. While you cannot add entire clusters to your categories, you can add the concepts in a cluster to a category through the Cluster Definitions dialog box. See the topic [Cluster Definitions](#) for more information.

You can make changes to the settings for clustering to influence the results. See the topic [Building Clusters](#) for more information.

Figure 1. Clusters view



The Clusters view is organized into three panes, each of which can be hidden or shown by selecting its name from the View menu. Typically, only the Clusters pane and the Visualization pane are visible.

Clusters Pane

Located on the left side, this pane presents the clusters that were discovered in the text data. You can create clustering results by clicking the Build button. Clusters are formed by a clustering algorithm, which attempts to identify concepts that occur together frequently.

Whenever a new extraction takes place, the cluster results are cleared, and you have to rebuild the clusters to get the latest results. When building the clusters, you can change some settings, such as the maximum number of clusters to create, the maximum number of concepts it can contain, or the maximum number of links with external concepts it can have. See the topic [Exploring Clusters](#) for more information.

Visualization Pane

Located in the upper right corner, this pane offers two perspectives on clustering: a Concept Web graph and a Cluster Web graph. If not visible, you can access this pane from the View menu (View > Visualization). Depending on what is selected in the clusters pane, you can view the corresponding interactions between or within clusters. The results are presented in multiple formats:

- Concept Web. Web graph showing all of the concepts within the selected cluster(s), as well as linked concepts outside the cluster.
- Cluster Web. Web graph showing the links from the selected cluster(s) to other clusters, as well as any links between those other clusters.

Note: In order to display a Cluster Web graph, you must have already built clusters with external links. External links are links between concept pairs in separate clusters (a concept within one cluster and a concept outside in another cluster). See the topic [Cluster Graphs](#) for more information.

Data Pane

The Data pane is located in the lower right corner and is hidden by default. You cannot display any Data pane results from the Clusters pane since these clusters span multiple documents/records, making the data results uninteresting. However, you can see the data corresponding to a selection within the Cluster Definitions dialog box. Depending on what is selected in that dialog box, only the corresponding text appears in the Data pane. Once you make a selection, click the Display & button to populate the Data pane with the documents or records that contain all of the concepts together.

The corresponding documents or records show the concepts highlighted in color to help you easily identify them in the text. You can also hover your mouse over color-coded items to display the concept under which it was extracted and the type to which it was assigned. The Data pane can contain multiple columns but the text field column is always shown. It carries the name of the text field that was used during extraction or a document name if the text data is in many different files. Other columns are available. See the topic [The Data Pane](#) for more information.

The Text Link Analysis view

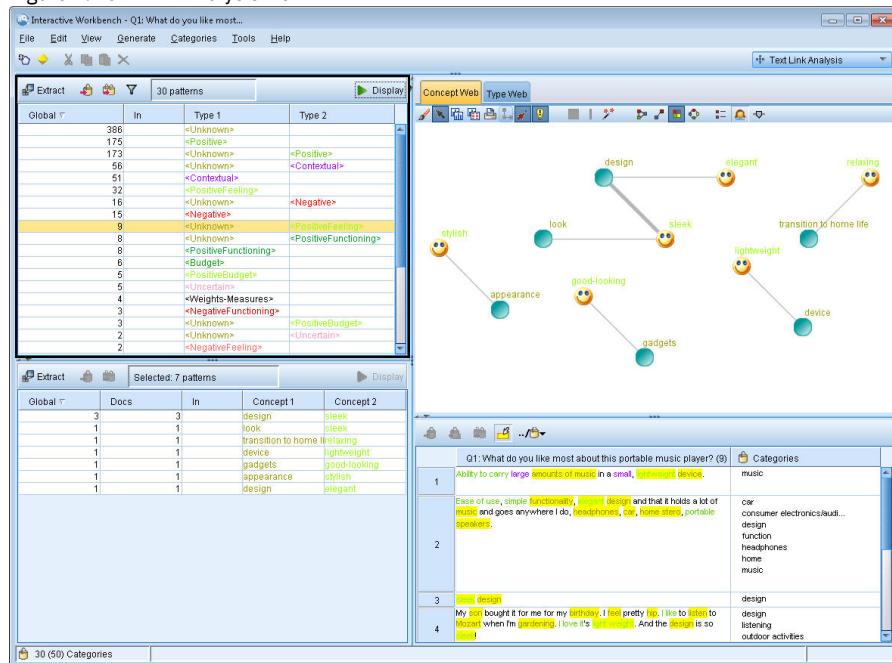
In the Text Link Analysis view, you can build and explore text link analysis patterns found in your text data. Text link analysis (TLA) is a pattern-matching technology that enables you to define TLA rules and compare them to actual extracted concepts and relationships found in your text.

Patterns are most useful when you are attempting to discover relationships between concepts or opinions about a particular subject. Some examples include wanting to extract opinions on products from survey data, genomic relationships from within medical research papers, or relationships between people or places from intelligence data.

Once you've extracted some TLA patterns, you can explore them in the Data or Visualization panes and even add them to categories in the Categories and Concepts view. There must be some TLA rules defined in the resource template or libraries you are using in order to extract TLA results. See the topic [About Text Link Rules](#) for more information.

If you chose to extract TLA pattern results, the results are presented in this view. If you have not chosen to do so, you will have to use the Extract button and choose the option to enable the extraction of patterns.

Figure 1. Text Link Analysis view



The Text Link Analysis view is organized into four panes, each of which can be hidden or shown by selecting its name from the View menu. See the topic [Exploring Text Link Analysis](#) for more information.

Type and Concept Patterns Panes

Located on the left side, the Type and Concept Pattern panes are two interconnected panes in which you can explore and select your TLA pattern results. Patterns are made up of a series of up to either six types or six concepts. The TLA pattern rule as it is defined in the linguistic resources dictates the complexity of the pattern results. See the topic [About Text Link Rules](#) for more information.

Pattern results are first grouped at the type level and then divided into concept patterns. For this reason, there are two different result panes: Type Patterns (upper left) and Concept Patterns (lower left).

- Type Patterns. The Type Patterns pane presents extracted patterns consisting of two or more related types matching a TLA pattern rule. Type patterns are shown as **<Organization> + <Location> + <Positive>**, which might provide positive feedback about an organization in a specific location.
- Concept Patterns. The Concept Patterns pane presents the extracted patterns at the concept level for all of the type pattern(s) currently selected in the Type Patterns pane above it. Concept patterns follow a structure such as **hotel + paris + wonderful**.

Just as with the extraction results in the Categories and Concepts view, you can review the results here. If you see any refinements you would like to make to the types and concepts that make up these patterns, you make those in the Extraction Results pane in the Categories and Concepts view, or directly in the Resource Editor, and reextract your patterns.

Visualization Pane

Located in the upper right corner of the Text Link Analysis view, this pane presents a web graph of the selected patterns as either type patterns or concept patterns. If not visible, you can access this pane from the View menu (View > Visualization). Depending on what is selected in the other panes, you can view the corresponding interactions between documents/records and the patterns.

The results are presented in multiple formats:

- Concept Graph. This graph presents all the concepts in the selected pattern(s). The line width and node sizes (if type icons are not shown) in a concept graph show the number of global occurrences in the selected table.
- Type Graph. This graph presents all the types in the selected pattern(s). The line width and node sizes (if type icons are not shown) in the graph show the number of global occurrences in the selected table. Nodes are represented by either a type color or by an icon.

See the topic [Text Link Analysis Graphs](#) for more information.

Data Pane

The Data pane is located in the lower right corner. This pane presents a table containing the documents or records corresponding to a selection in another area of the view. Depending on what is selected, only the corresponding text appears in the Data pane. Once you make a selection, click a Display button to populate the Data pane with the corresponding text.

If you have a selection in another pane, the corresponding documents or records show the concepts highlighted in color to help you easily identify them in the text. You can also hover your mouse over color-coded items to display a tooltip showing name of the concept under which it was extracted and the type to which it was assigned. See the topic [The Data Pane](#) for more information.

The Resource Editor view

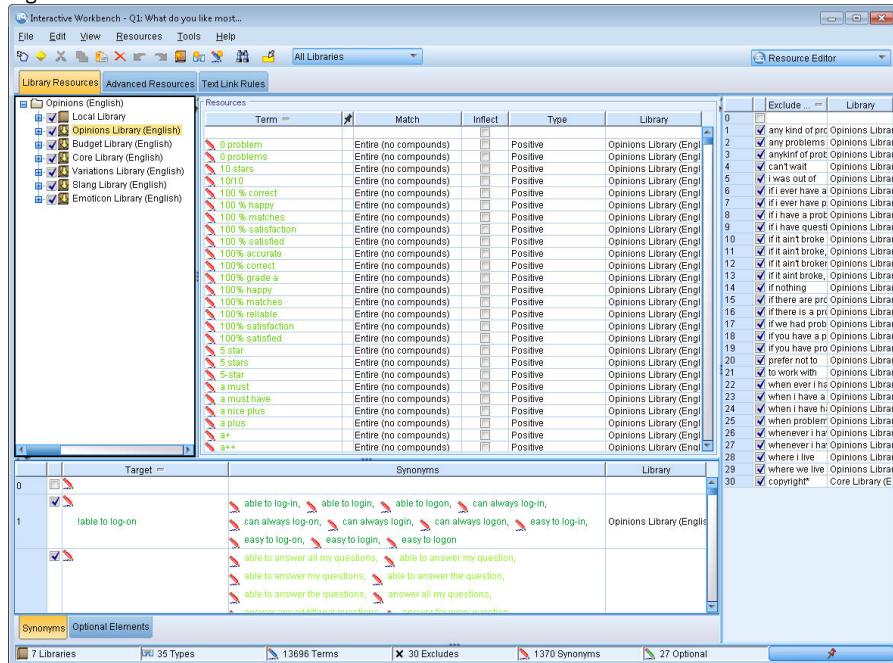
IBM® SPSS® Modeler Text Analytics rapidly and accurately captures key concepts from text data using a robust extraction engine. This engine relies heavily on linguistic resources to dictate how large amounts of unstructured, textual data should be analyzed and interpreted.

The Resource Editor view is where you can view and fine-tune the linguistic resources used to extract concepts, group them under types, discover patterns in the text data, and much more. IBM SPSS Modeler Text Analytics offers several preconfigured resource templates. Also, in some languages, you can also use the resources in a text analysis packages. See the topic [Using Text Analysis Packages](#) for more information.

Since these resources may not always be perfectly adapted to the context of your data, you can create, edit, and manage your own resources for a particular context or domain in the Resource Editor. See the topic [Working with Libraries](#) for more information.

To simplify the process of fine-tuning your linguistic resources, you can perform common dictionary tasks directly from the Categories and Concepts view through context menus in the Extraction Results and Data panes. See the topic [Refining extraction results](#) for more information.

Figure 1. Resource Editor view



The operations that you perform in the Resource Editor view revolve around the management and fine-tuning of the linguistic resources. These resources are stored in the form of templates and libraries. The Resource Editor view is organized into four parts: Library Tree pane, Type Dictionary pane, Substitution Dictionary pane, and Exclude Dictionary pane.

Note: See the topic [The Editor interface](#) for more information.

Setting Options

You can set general options for IBM® SPSS® Modeler Text Analytics in the Options dialog box. This dialog box contains the following tabs:

- **Session.** This tab contains general options and delimiters. See the topic [Options: Session Tab](#) for more information.
- **Display.** This tab contains options for the colors used in the interface. See the topic [Options: Display Tab](#) for more information.
- **Sounds.** This tab contains options for sound cues. See the topic [Options: Sounds Tab](#) for more information.

To Edit Options

1. From the menus, choose Tools > Options. The Options dialog box opens.
 2. Select the tab containing the information you want to change.
 3. Change any of the options.
 4. Click OK to save the changes.
- [Options: Session Tab](#)
 - [Options: Display Tab](#)
 - [Options: Sounds Tab](#)

Related information

- [Interactive workbench mode](#)
- [The Categories and Concepts View](#)
- [The Clusters View](#)
- [The Text Link Analysis view](#)
- [The Resource Editor view](#)
- [Generating Model Nuggets and Modeling Nodes](#)
- [Updating Modeling Nodes and Saving](#)
- [Closing and Ending Sessions](#)
- [Keyboard Accessibility](#)
- [Options: Session Tab](#)
- [Options: Display Tab](#)
- [Options: Sounds Tab](#)
- [Microsoft Internet Explorer settings for Help](#)

Options: Session Tab

On this tab, you can define some of the basic settings.

Data Pane and Category Graph Display. These options affect how data are presented in the Data pane and in the Visualization pane in the Categories and Concepts view.

- **Display limit for Data pane and Category Web.** This option sets the maximum number of documents to show or use to populate the Data panes or graphs and charts in the Categories and Concepts view.
- **Show categories for documents/records at Display time.** If selected, the documents or records are scored whenever you click Display so that any categories to which they belong can be displayed in the Categories column in the Data pane as well as in the category graphs. In some cases, especially with larger datasets, you may want to turn off this option so that data and graphs are displayed much faster.

Add to Category from Data Pane. These options affect what is added to categories when documents and records are added from the Data pane.

- **In Categories and Concepts view, copy.** Adding a document or record from the Data pane in this view will copy over either Concepts only or both Concepts and Patterns.
- **In Text Link Analysis view, copy.** Adding a document or record from the Data pane in this view will copy over either Patterns only or both Concepts and Patterns.

Resource Editor delimiter. Select the character to be used as a delimiter when entering elements, such as concepts, synonyms, and optional elements, in the Resource Editor view.

Related information

- [Setting Options](#)
- [Options: Display Tab](#)
- [Options: Sounds Tab](#)
- [Microsoft Internet Explorer settings for Help](#)

Options: Display Tab

On this tab, you can edit options affecting the overall look and feel of the application and the colors used to distinguish elements.

Note: To switch the look and feel of the product to a classic look or one from a previous release, open the User Options dialog in the Tools menu in the main IBM® SPSS® Modeler window.

Custom Colors. Edit the colors for elements appearing onscreen. For each of the elements in the table, you can change the color. To specify a custom color, click the color area to the right of the element you want to change and choose a color from the drop-down color list.

- **Non-extracted text.** Text data that was not extracted yet visible in the Data pane.
- **Highlight background.** Text selection background color when selecting elements in the panes or text in the Data pane.
- **Extraction needed background.** Background color of the Extraction Results, Patterns, and Clusters panes indicating that changes have been made to the libraries and an extraction is needed.
- **Category feedback background.** Category background color that appears after an operation.
- **Default type.** Default color for types and concepts appearing in the Data pane and Extraction Results pane. This color will apply to any custom types that you create in the Resource Editor. You can override this default color for your custom type dictionaries by editing the properties for these type dictionaries in the Resource Editor. See the topic [Creating types](#) for more information.
- **Striped table 1.** First of the two colors used in an alternating manner in the table in the Edit Forced concepts dialog box in order to differentiate each set of lines.
- **Striped table 2.** Second of the two colors used in an alternating manner in the table in the Edit Forced concepts dialog box in order to differentiate each set of lines.

Note: If you click the Reset to Defaults button, all options in this dialog box are reset to the values they had when you first installed this product.

Related information

- [Setting Options](#)
 - [Options: Session Tab](#)
 - [Options: Sounds Tab](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Options: Sounds Tab

On this tab, you can edit options affecting sounds. Under Sound Events, you can specify a sound to be used to notify you when an event occurs. A number of sounds are available. Use the ellipsis button (...) to browse for and select a sound. The .wav files used to create sounds for IBM® SPSS® Modeler Text Analytics are stored in the *media* subdirectory of the installation directory. If you do not want sounds to be played, select Mute All Sounds. Sounds are muted by default.

Note: If you click the Reset to Defaults button, all options in this dialog box are reset to the values they had when you first installed this product.

Related information

- [Setting Options](#)
 - [Options: Session Tab](#)
 - [Options: Display Tab](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Microsoft Internet Explorer settings for Help

Microsoft Internet Explorer Settings

Most Help features in this application use technology based on Microsoft Internet Explorer. Some versions of Internet Explorer (including the version provided with Microsoft Windows XP, Service Pack 2) will by default block what it considers to be "active content" in Internet Explorer windows on your local computer. This default setting may result in some blocked content in Help features. To see all Help content, you can change the default behavior of Internet Explorer.

1. From the Internet Explorer menus choose:
Tools > Internet Options...
2. Click the Advanced tab.
3. Scroll down to the Security section.
4. Select Allow active content to run in files on My Computer.

Generating Model Nuggets and Modeling Nodes

When you are in an interactive session, you may want to use the work you have done to generate either:

- **A text mining modeling node.** A modeling node generated from an interactive workbench session is a Text Mining node whose settings and options reflect those stored in the open interactive session. This can be useful when you no longer have the original Text Mining node or when you want to make a new version. See the topic [Mining for Concepts and Categories](#) for more information.
- **A category model nugget.** A model nugget generated from an interactive workbench session is a category model nugget. You must have at least one category in the Categories and Concepts view in order to generate a category model nugget. See the topic [Text Mining Nugget: Category Model](#) for more information.

To Generate a Text Mining Modeling Node

1. From the menus, choose Generate > Generate Modeling Node. A Text Mining modeling node is added to the working canvas using all of the settings currently in the workbench session. The node is named after the text field.

To Generate a Category Model Nugget

1. From the menus, choose Generate > Generate Model. A model nugget is generated directly onto the Model palette with the default name.

Related information

- [Interactive workbench mode](#)
- [The Categories and Concepts View](#)
- [The Clusters View](#)
- [The Text Link Analysis view](#)
- [The Resource Editor view](#)
- [Setting Options](#)
- [Updating Modeling Nodes and Saving](#)
- [Closing and Ending Sessions](#)
- [Keyboard Accessibility](#)
- [Microsoft Internet Explorer settings for Help](#)

Updating Modeling Nodes and Saving

While you are working in an interactive session, we recommend that you update the modeling node from time to time to save your changes. You should also update your modeling node whenever you are finished working in the interactive workbench session and want to save your work. When you update the modeling node, the workbench session content is saved back to the Text Mining node that originated the interactive workbench session. This does not close the output window.

Important! This update will not save your stream. To save your stream, do so in the main IBM® SPSS® Modeler window after updating the modeling node.

To Update a Modeling Node

1. From the menus, choose File > Update Modeling Node. The modeling node is updated with the build and extraction settings, along with any options and categories you have.

Related information

- [Interactive workbench mode](#)
- [The Categories and Concepts View](#)
- [The Clusters View](#)
- [The Text Link Analysis view](#)
- [The Resource Editor view](#)
- [Setting Options](#)
- [Generating Model Nuggets and Modeling Nodes](#)
- [Closing and Ending Sessions](#)
- [Keyboard Accessibility](#)
- [Microsoft Internet Explorer settings for Help](#)

Closing and Ending Sessions

When you are finished working in your session, you can leave the session in three different ways:

- **Save.** This option allows you to first save your work back into the originating modeling node for future sessions, as well as to publish any libraries for reuse in other sessions. See the topic [Sharing Libraries](#) for more information. After you have saved, the session window is closed, and the session is deleted from the Output manager in the IBM® SPSS® Modeler window.
- **Exit.** This option will discard any unsaved work, close the session window, and delete the session from the Output manager in the IBM SPSS Modeler window. To free up memory, we recommend saving any important work and exiting the session.
- **Close.** This option will not save or discard any work. This option closes the session window but the session will continue to run. You can open the session window again by selecting this session in the Output manager in the IBM SPSS Modeler window.

To Close a Workbench Session

1. From the menus, choose File > Close.

Related information

- [Interactive workbench mode](#)
 - [The Categories and Concepts View](#)
 - [The Clusters View](#)
 - [The Text Link Analysis view](#)
 - [The Resource Editor view](#)
 - [Setting Options](#)
 - [Generating Model Nuggets and Modeling Nodes](#)
 - [Updating Modeling Nodes and Saving](#)
 - [Keyboard Accessibility](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Keyboard Accessibility

The interactive workbench interface offers keyboard shortcuts to make the product's functionality more accessible. At the most basic level, you can press the Alt key plus the appropriate key to activate window menus (for example, Alt+F to access the File menu) or press the Tab key to scroll through dialog box controls. This section will cover the keyboard shortcuts for alternative navigation. There are other keyboard shortcuts for the IBM® SPSS® Modeler interface.

Table 1. Generic keyboard shortcuts

Shortcut key	Function
Ctrl+1	Display the first tab in a pane with tabs.
Ctrl+2	Display the second tab in a pane with tabs.
Ctrl+A	Select all elements for the pane that has focus.
Ctrl+C	Copy selected text to the clipboard.
Ctrl+E	Launch extraction in Categories and Concepts and Text Link Analysis views.
Ctrl+F	Display the Find toolbar in the Resource Editor/Template Editor, if not already visible, and put focus there.
Ctrl+I	In the Categories and Concepts view, launch the Category Definitions dialog box for the selected category. In the Cluster view, launch the Cluster Definitions dialog box for the selected cluster.
Ctrl+R	Open the Add Terms dialog box in the Resource Editor/Template Editor.
Ctrl+T	Open the Type Properties dialog box to create a new type in the Resource Editor/Template Editor.
Ctrl+V	Paste clipboard contents.
Ctrl+X	Cut selected items from the Resource Editor/Template Editor.
Ctrl+Y	Redo the last action in the view.
Ctrl+Z	Undo the last action in the view.
F1	Display Help, or when in a dialog box, display context Help for an item.
F2	Toggle in and out of edit mode in table cells.
F6	Move the focus between the main panes in the active view.
F8	Move the focus to pane splitter bars for resizing.
F10	Expand the main File menu.
up arrow, down arrow	Resize the pane vertically when the splitter bar is selected.
left arrow, right arrow	Resize the pane horizontally when the splitter bar is selected.
Home, End	Resize panes to minimum or maximum size when the splitter bar is selected.
Tab	Move forward through items in the window, pane, or dialog box.
Shift+F10	Display the context menu for an item.

Shortcut key	Function
Shift+Tab	Move back through items in the window or dialog box.
Shift+arrow	Select characters in the edit field when in edit mode (F2).
Ctrl+Tab	Move the focus forward to the next main area in the window.
Shift+Ctrl+Tab	Move the focus backward to the previous main area in the window.

- [Shortcuts for Dialog Boxes](#)

Related information

- [Interactive workbench mode](#)
- [The Categories and Concepts View](#)
- [The Clusters View](#)
- [The Text Link Analysis view](#)
- [The Resource Editor view](#)
- [Setting Options](#)
- [Generating Model Nuggets and Modeling Nodes](#)
- [Updating Modeling Nodes and Saving](#)
- [Closing and Ending Sessions](#)
- [Shortcuts for Dialog Boxes](#)
- [Microsoft Internet Explorer settings for Help](#)

Shortcuts for Dialog Boxes

Several shortcut and screen reader keys are helpful when you are working with dialog boxes. Upon entering a dialog box, you may need to press the Tab key to put the focus on the first control and to initiate the screen reader. A complete list of special keyboard and screen reader shortcuts is provided in the following table.

Table 1. Dialog box shortcuts

Shortcut key	Function
Tab	Move forward through the items in the window or dialog box.
Ctrl+Tab	Move forward from a text box to the next item.
Shift+Tab	Move back through items in the window or dialog box.
Shift+Ctrl+Tab	Move back from a text box to the previous item.
space bar	Select the control or button that has focus.
Esc	Cancel changes and close the dialog box.
Enter	Validate changes and close the dialog box (equivalent to the OK button). If you are in a text box, you must first press Ctrl+Tab to exit the text box.

Related information

- [Keyboard Accessibility](#)
- [Microsoft Internet Explorer settings for Help](#)

Extracting Concepts and Types

Whenever you run a stream that launches the interactive workbench, an extraction is automatically performed on the text data in the stream. The end result of this extraction is a set of concepts, types, and, in the case where TLA patterns exist in the linguistic resources, patterns. You can view and work with concepts and types in the Extraction Results pane. See [How extraction works](#) for more information.

If you want to fine-tune the extraction results, you can modify the linguistic resources and re-extract. See [Refining extraction results](#) for more information. The extraction process relies on the resources and any parameters in the Extract dialog box to dictate how to extract and organize the results. You can use the extraction results to define the better part, if not all, of your category definitions.

Note: Beginning with version 18.2, extracted concept results have been improved (they're now similar to extracted concept results in IBM® SPSS® Text Analytics for Surveys).

- [Extraction results: Concepts and types](#)
- [Extracting data](#)
- [Filtering Extraction Results](#)

- [Exploring concept maps](#)
 - [Refining extraction results](#)
-

Extraction results: Concepts and types

During the extraction process, all of the text data is scanned and the relevant concepts are identified, extracted, and assigned to types. When the extraction is complete, the results appear in the Extraction Results pane located in the lower left corner of the Categories and Concepts view. The first time you launch the session, the linguistic resource template you selected in the node is used to extract and organize these concepts and types.

Note: If there are more results that can fit in the visible pane, you can use the controls at the bottom of the pane to move forwards and backwards through the results, or enter a page number to go to.

The concepts, types, and TLA patterns that are extracted are collectively referred to as **extraction results**, and they serve as the descriptors, or building blocks, for your categories. You can also use concepts, types, and patterns in your category rules. Additionally, the automatic techniques use concepts and types to build the categories.

Text mining is an iterative process in which extraction results are reviewed according to the context of the text data, fine-tuned to produce new results, and then reevaluated. After extracting, you should review the results and make any changes that you find necessary by modifying the linguistic resources. You can fine-tune the resources, in part, directly from the Extraction Results pane, Data pane, Category Definitions dialog box, or Cluster Definitions dialog box. See the topic [Refining extraction results](#) for more information. You can also do so directly in the Resource Editor view. See the topic [The Resource Editor view](#) for more information.

After fine-tuning, you can then reextract to see the new results. By fine-tuning your extraction results from the start, you can be assured that each time you reextract, you will get identical results in your category definitions, perfectly adapted to the context of the data. In this way, documents/records will be assigned to your category definitions in a more accurate, repeatable manner.

Concepts

During the extraction process, the text data is scanned and analyzed in order to identify interesting or relevant single words (such as **election** or **peace**) and word phrases (such as **presidential**

election, election of the president, or peace treaties) in the text. These words and phrases are collectively referred to as **terms**. Using the linguistic resources, the relevant terms are extracted and then similar terms are grouped together under a lead term called a **concept**.

You can see the set of underlying terms for a concept by hovering your mouse over the concept name. Doing so will display a tooltip showing the concept name and up to several lines of terms that are grouped under that concept. These underlying terms include the synonyms defined in the linguistic resources (regardless of whether they were found in the text or not) as well as the any extracted plural/singular terms, permuted terms, terms from fuzzy grouping, and so on. You can copy these terms or see the full set of underlying terms by right-clicking the concept name and choosing the context menu option.

By default, the concepts are shown in lowercase and sorted in descending order according to the document count (Doc. column). When concepts are extracted, they are assigned a type to help group similar concepts. They are color coded according to this type. Colors are defined in the type properties within the Resource Editor. See the topic [Type dictionaries](#) for more information.

Whenever a concept, type, or pattern is being used in a category definition, an icon appears in the sortable In column.

Types

Types are semantic groupings of concepts. When concepts are extracted, they are assigned a type to help group similar concepts. Several built-in types are delivered with IBM® SPSS® Modeler Text Analytics , such as **<Location>**, **<Organization>**, **<Person>**, **<Positive>**, **<Negative>** and so on. For example, the **<Location>** type groups geographical keywords and places. This type would be assigned to concepts such as **chicago**, **paris**, and **tokyo**. For most languages, concepts that are not found in any type dictionary but are extracted from the text are automatically typed as **<Unknown>** See the topic [Built-in types](#) for more information.

When you select the Type view, the extracted types appear by default in descending order by global frequency. You can also see that types are color coded to help distinguish them. Colors are part of the type properties. See the topic [Creating types](#) for more information. You can also create your own types.

Patterns

Patterns can also be extracted from your text data. However, you must have a library that contains some Text Link Analysis (TLA) pattern rules in the Resource Editor. You also have to choose to extract these patterns in the IBM SPSS Modeler Text Analytics node setting or in the Extract dialog box using the option Enable Text Link Analysis pattern extraction. See the topic [Exploring Text Link Analysis](#) for more information.

Extracting data

Whenever an extraction is needed, the Extraction Results pane becomes yellow in color and the message Press Extract Button to Extract Concepts appears below the toolbar in this pane.

You may need to extract if you do not have any extraction results yet, have made changes to the linguistic resources and need to update the extraction results, or have reopened a session in which you did not save the extraction results (Tools > Options).

Note: If you change the source node for your stream after extraction results have been cached with the Use session work... option, you will need to run a new extraction once the interactive workbench session is launched if you want to get updated extraction results.

When you run an extraction, a progress indicator appears to provide feedback on the status of the extraction. During this time, the extraction engine reads through all of the text data and identifies the relevant terms and patterns and extracts them and assigns them to a type. Then, the engine attempts groups synonyms terms under one lead term, called a concept. When the process is complete, the resulting concepts, types, and patterns appear in the Extraction Results pane.

The extraction process results in a set of concepts and types, as well as Text Link Analysis (TLA) patterns, if enabled. You can view and work with these concepts and types in the Extraction Results pane in the Categories and Concepts view. If you extracted TLA patterns, you can see those in the Text Link Analysis view.

Note: There is a relationship between the size of your dataset and the time it takes to complete the extraction process. You can always consider inserting a Sample node upstream or optimizing your machine's configuration.

To extract data

1. From the menus, choose Tools > Extract. Alternatively, click the Extract toolbar button.
2. If you chose to always display the Extraction Settings dialog, it appears so that you can make any changes. See further in this topic for descriptors of each settings.
3. Click Extract to begin the extraction process. Once the extraction begins, the progress dialog box opens. After extraction, the results appear in the Extraction Results pane. By default, the concepts are shown in lowercase and sorted in descending order according to the document count (Doc. column).

You can review the results using the toolbar options to sort the results differently, to filter the results, or to switch to a different view (concepts or types). You can also refine your extraction results by working with the linguistic resources. See the topic [Refining extraction results](#) for more information.

Potential extraction issues

Multiple Interactive Workbench sessions can cause sluggish behavior. SPSS® Modeler Text Analytics and SPSS Modeler share a common Java run-time engine when an interactive workbench session is launched. Depending on the number of Interactive Workbench sessions you invoke during a SPSS Modeler session - even if opening and closing the same session - system memory may cause the application to become sluggish. This effect may be especially pronounced if you are working with large data or have a machine with less than the recommended RAM setting of 4GB. If you notice your machine is slow to respond, it is recommended that you save all your work, shut down SPSS Modeler, and re-launch the application. Running SPSS Modeler Text Analytics on a machine with less than the recommended memory, particularly when working with large data sets or for prolonged periods of time, may cause Java to run out of memory and shut down. It is strongly suggested you upgrade to the recommended memory setting or larger (or use SPSS Modeler Text Analytics Server) if you work with large data.

For Dutch, English, French, German, Italian, Portuguese, and Spanish Text

The Extraction Settings dialog box contains some basic extraction options.

Enable Text Link Analysis pattern extraction. Specifies that you want to extract TLA patterns from your text data. It also assumes you have TLA pattern rules in one of your libraries in the Resource Editor. This option may significantly lengthen the extraction time. See the topic [Exploring Text Link Analysis](#) for more information.

Accommodate punctuation errors. This option temporarily normalizes text containing punctuation errors (for example, improper usage) during extraction to improve the extractability of concepts. This option is extremely useful when text is short and of poor quality (as, for example, in open-ended survey responses, e-mail, and CRM data), or when the text contains many abbreviations.

Accommodate spelling for a minimum word character length of [n] This option applies a fuzzy grouping technique that helps group commonly misspelled words or closely spelled words under one concept. The fuzzy grouping algorithm temporarily strips all vowels (except the first one) and strips double/triple consonants from extracted words and then compares them to see if they are the same so that **modeling** and **modelling** would be grouped together. However, if each term is assigned to a different type, excluding the **<Unknown>** type, the fuzzy grouping technique will not be applied.

You can also define the minimum number of *root* characters required before fuzzy grouping is used. The number of root characters in a term is calculated by totaling all of the characters and subtracting any characters that form inflection suffixes and, in the case of compound-word terms, determiners and prepositions. For example, the term **exercises** would be counted as 8 root characters in the form "exercise," since the letter **s** at the end of the word is an inflection (plural form). Similarly, **apple sauce** counts as 10 root characters ("apple sauce") and **manufacturing of cars** counts as 16 root characters ("manufacturing car"). This method of counting is only used to check whether the fuzzy grouping should be applied but does not influence how the words are matched.

Note: If you find that certain words are later grouped incorrectly, you can exclude word pairs from this technique by explicitly declaring them in the Fuzzy Grouping: Exceptions section in the Advanced Resources tab. See the topic [Fuzzy Grouping](#) for more information.

Extract uniterms This option extracts single words (uniterms) as long as the word is not already part of a compound word and if it is either a noun or an unrecognized part of speech.

Extract nonlinguistic entities This option extracts nonlinguistic entities, such as phone numbers, social security numbers, times, dates, currencies, digits, percentages, e-mail addresses, and HTTP addresses. You can include or exclude certain types of nonlinguistic entities in the Nonlinguistic Entities: Configuration section of the Advanced Resources tab. By disabling any unnecessary entities, the extraction engine won't waste processing time. See the topic [Configuration](#) for more information.

Uppercase algorithm This option extracts simple and compound terms that are not in the built-in dictionaries as long as the first letter of the term is in uppercase. This option offers a good way to extract most proper nouns.

Group partial and full person names together when possible This option groups names that appear differently in the text together. This feature is helpful since names are often referred to in their full form at the beginning of the text and then only by a shorter version. This option attempts to match any uniterm with the <Unknown> type to the last word of any of the compound terms that is typed as <Person>. For example, if *doe* is found and initially typed as <Unknown>, the extraction engine checks to see if any compound terms in the <Person> type include *doe* as the last word, such as *john doe*. This option does not apply to first names since most are never extracted as uniterms.

Maximum nonfunction word permutation This option specifies the maximum number of nonfunction words that can be present when applying the permutation technique. This permutation technique groups similar phrases that differ from each other only by the nonfunction words (for example, *of* and *the*) contained, regardless of inflection. For example, let's say that you set this value to at most two words and both **company officials** and **officials of the company** were extracted. In this case, both extracted terms would be grouped together in the final concept list since both terms are deemed to be the same when *of the* is ignored.

Use derivation when grouping multiterms When processing Big Data, select this option to group multiterms by using derivation rules.

Index Option for Concept Map Specifies that you want to build the map index at extraction time so that concept maps can be drawn quickly later. To edit the index settings, click Settings. See the topic [Building Concept Map Indexes](#) for more information.

Always show this dialog before starting an extraction Specify whether you want to see the Extraction Settings dialog each time you extract, if you never want to see it unless you go to the Tools menu, or whether you want to be asked each time you extract if you want to edit any extraction settings.

Filtering Extraction Results

When you are working with very large datasets, the extraction process could produce millions of results. For many users, this amount can make it more difficult to review the results effectively. Therefore, in order to zoom in on those that are most interesting, you can filter these results through the Filter dialog available in the Extraction Results pane.

Keep in mind that all of the settings in this Filter dialog are used together to filter the extraction results that are available for categories.

Filter by Frequency You can filter to display only those results with a certain global or document frequency value.

- Global frequency is the total number of times a concept appears in the entire set of documents or records and is shown in the Global column.
- Document frequency is the total number of documents or records in which a concept appears and is shown in the Docs column.

For example, if the concept **nato** appeared 800 times in 500 records, we would say that this concept has a global frequency of 800 and a document frequency of 500.

And by Type You can filter to display only those results belonging to certain types. You can choose all types or only specific types.

And by Match Text You can also filter to display only those results that match the rule you define here. Enter the set of characters to be matched in the Match text field and then select the condition in which to apply the match.

Table 1. Match text conditions

Condition	Description
Contains	The text is matched if the string occurs anywhere. (Default choice)
Starts with	Text is matched only if the concept or type starts with the specified text.
Ends with	Text is matched only if the concept or type ends with the specified text.
Exact match	The entire string must match the concept or type name.

Results Displayed in Extraction Result Pane

Here are some examples of how the results might be displayed, in English, in the Extraction Results pane toolbar based on the filters.

Table 2. Examples of filter feedback

Filter feedback	Description
404 concepts	The toolbar shows the number of results. Since there was no text matching filter and the maximum was not met, no additional icons are shown.
358 concepts !	The toolbar shows results were limited to the maximum specified in the filter, which in this case was 300. If a purple icon is present, this means that the maximum number of concepts was met. Hover over the icon for more information.
4 concepts !	The toolbar shows results were limited using a match text filter. This is shown by the magnifying glass icon.

To Filter the Results

- From the menus, choose Tools > Filter. The Filter dialog box opens.
- Select and refine the filters you want to use.
- Click OK to apply the filters and see the new results in the Extraction Results pane.

Related information

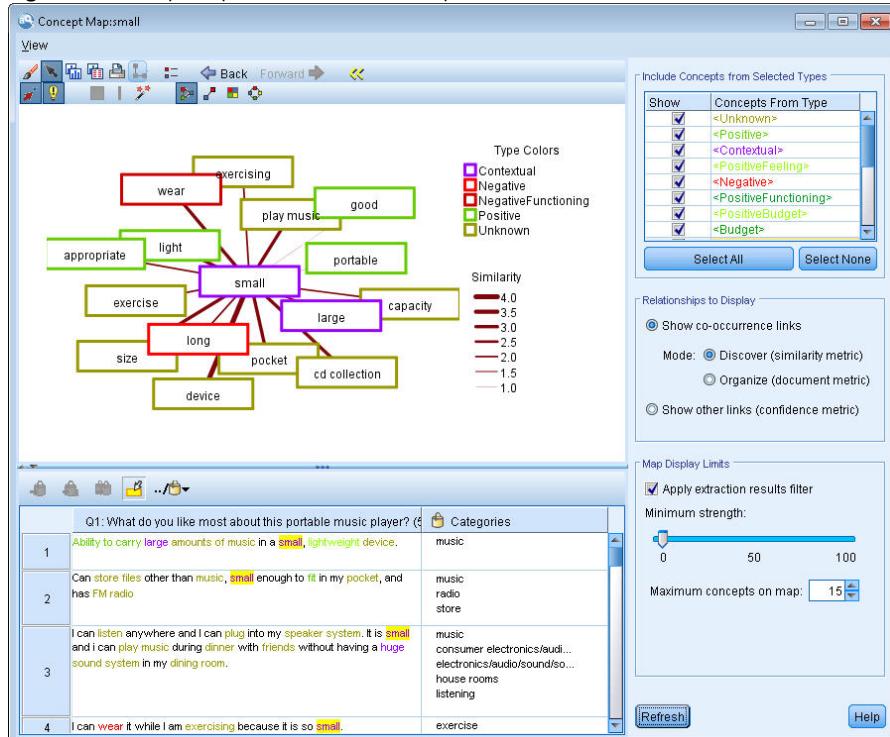
- [Extracting Concepts and Types](#)
- [Extraction results: Concepts and types](#)
- [Extracting data](#)
- [Exploring concept maps](#)
- [Building Concept Map Indexes](#)
- [Refining extraction results](#)
- [Microsoft Internet Explorer settings for Help](#)

Exploring concept maps

You can create a concept map to explore how concepts are interrelated. By selecting a single concept and clicking Map, a concept map window opens so that you can explore the set of concepts that are related to the selected concept. You can filter out which concepts are displayed by editing the settings such as which types to include, what kinds of relationship to look for and so on.

Important: Before a map can be created, an index must be generated. This may take several minutes. However, once you have generated the index, you do not have regenerate it again until you re-extract. If you want the index to be generated automatically each time you extract, select that option in the extraction settings. See the topic [Extracting data](#) for more information.

Figure 1. A concept map for the selected concept



To View a Concept Map

1. In the Extraction Results pane, select a single concept.
2. In the toolbar of this pane, click the Map button. If the map index was already generated the concept map opens in a separate dialog. If the map index was not generated or was out of date, the index must be rebuilt. This process may take several minutes.
3. Click around the map to explore. If you double-click a linked concept, the map will redraw itself and show you the linked concepts for the concept you just double-clicked.
4. The top toolbar offers some basic map tools such as moving back to a previous map, filtering links according to relationship strengths, and also opening the filter dialog to control the types of concepts that appear as well as the kinds of relationships to represent. A second toolbar line contains graph editing tools. See the topic [Using Graph Toolbars and Palettes](#) for more information.
5. If you are unsatisfied with the kinds of links being found, review the settings for this map show on the right side of the map.

Map Settings: Include Concepts from Selected Types

Only those concepts belonging to the selected types in the table are shown in the map. To hide concepts from a certain type, deselect that type in the table.

Map Settings: Relationships to Display

Show co-occurrence links If you want to show co-occurrence links, choose the mode. The mode affects how the link strength was calculated.

- *Discover (similarity metric)*. With this metric, the strength of the link is calculated using more complex calculation that takes into account how often two concepts appear apart as well as how often they appear together. A high strength value means that a pair of concepts tend to appear more frequently together than to appear apart. With the following formula, any floating point values are converted to integers.

Figure 2. Similarity coefficient formula

$$\text{similarity coefficient} = \frac{(C_{IJ})^2}{(C_I \times C_J)}$$

In this formula, C_I is the number of documents or records in which the concept I occurs.

C_J is the number of documents or records in which the concept J occurs.

C_{IJ} is the number of documents or records in which concept pair I and J co-occurs in the set of documents.

- *Organize (document metric)*. The strength of the links with this metric is determined by the raw count of co-occurrences. In general, the more frequent two concepts are, the more likely they are to occur together at times. A high strength value means that a pair of concepts appear together frequently.

Show other links (confidence metric). You can choose other links to display; these may be semantic, derivation (morphological), or inclusion (syntactical) and are related to how many steps removed a concept is from the concept to which it is linked. These can help you tune resources, particularly synonymy or to disambiguate. For short descriptions of each of these grouping techniques, see [Advanced linguistic settings](#)

Note: Keep in mind that if these were not selected when the index was built or if no relationships were found, then none will be displayed. See the topic [Building Concept Map Indexes](#) for more information.

Map Settings: Map Display Limits

Apply extraction results filter. If you do not want to use all of the concepts, you can use the filter in the extraction results pane to limit what is shown. Then select this option and IBM® SPSS® Modeler Text Analytics will look for related concepts using this filtered set. See the topic [Filtering Extraction Results](#) for more information.

Minimum strength. Set the minimum link strength here. Any related concepts with a relationship strength lower than this limit will be hidden from the map.

Maximum concepts on map. Specify the maximum number of relationships to show on the map.

- [Building Concept Map Indexes](#)

Building Concept Map Indexes

Before a map can be created, an index of concept relationships must be generated. Whenever you create a concept map, IBM® SPSS® Modeler Text Analytics refers to this index. You can choose which relationships to index by selecting the techniques in this dialog.

Grouping techniques. Choose one or more technique. For short descriptions of each of these techniques, see [About linguistic techniques](#) Not all techniques are available for all text languages.

Prevent pairing of specific concepts. Select this checkbox to stop the process from grouping or pairing two concepts together in the output. To create or manage concept pairs, click Manage Pairs. See the topic [Managing Link Exception Pairs](#) for more information.

Building the index may take several minutes. However, once you have generated the index, you do not have to regenerate it again until you re-extract or unless you want to change the settings to include more relationships. If you want to generate an index whenever you extract, you can select that option in the extraction settings. See the topic [Extracting data](#) for more information.

Related information

- [Extracting Concepts and Types](#)
 - [Extraction results: Concepts and types](#)
 - [Extracting data](#)
 - [Filtering Extraction Results](#)
 - [Exploring concept maps](#)
 - [Refining extraction results](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Refining extraction results

Extraction is an iterative process whereby you can extract, review the results, make changes to them, and then re-extract to update the results. Since accuracy and continuity are essential to successful text mining and categorization, fine-tuning your extraction results from the start ensures that each time you re-extract, you will get precisely the same results in your category definitions. In this way, records and documents will be assigned to your categories in a more accurate, repeatable manner.

The extraction results serve as the building blocks for categories. When you create categories using these extraction results, records and documents are automatically assigned to categories if they contain text that matches one or more category descriptors. Although you can begin categorizing before making any refinements to the linguistic resources, it is useful to review your extraction results at least once before beginning.

As you review your results, you may find elements that you want the extraction engine to handle differently. Consider the following examples:

- Unrecognized synonyms. Suppose you find several concepts you consider to be synonymous, such as **smart**, **intelligent**, **bright**, and **knowledgeable**, and they all appear as individual concepts in the extraction results. You could create a synonym definition in which **intelligent**, **bright**, and **knowledgeable** are all grouped under the target concept **smart**. Doing so would group all of these together with **smart**, and the global frequency count would be higher as well. See the topic [Adding synonyms](#) for more information.
- Mistyped concepts. Suppose that the concepts in your extraction results appear in one type and you would like them to be assigned to another. In another example, imagine that you find 15 vegetable concepts in your extraction results and you want them all to be added to a new type called **<Vegetable>**. For most languages, concepts that are not found in any type dictionary but are extracted from the text are automatically typed as **<Unknown>**. You can add concepts to types. See the topic [Adding concepts to types](#) for more information.
- Insignificant concepts. Suppose that you find a concept that was extracted and has a very high frequency count—that is, it is found in many records or documents. However, you consider this concept to be insignificant to your analysis. You can exclude it from extraction. See the topic [Excluding concepts from extraction](#) for more information.
- Incorrect matches. Suppose that in reviewing the records or documents that contain a certain concept, you discover that two words were incorrectly grouped together, such as **faculty** and **facility**. This match may be due to an internal algorithm, referred to as fuzzy grouping, that temporarily ignores double or triple consonants and vowels in order to group common misspellings. You can add these words to a list of word pairs that should not be grouped. See the topic [Fuzzy Grouping](#) for more information.
- Unextracted concepts. Suppose that you expect to find certain concepts extracted but notice that a few words or phrases were not extracted when you review the record or document text. Often these words are verbs or adjectives that you are not interested in. However, sometimes you do want to use a word or phrase that was not extracted as part of a category definition. To extract the concept, you can force a term into a type dictionary. See the topic [Forcing Words into Extraction](#) for more information.

Many of these changes can be performed directly from the Extraction Results pane, Data pane, Category Definitions dialog box, or Cluster Definitions dialog box by selecting one or more elements and right-clicking your mouse to access the context menus.

After making your changes, the pane background color changes to show that you need to re-extract to view your changes. See the topic [Extracting data](#) for more information. If you are working with larger data sets, it may be more efficient to re-extract after making several changes rather than after each change.

Note: You can view the entire set of editable linguistic resources used to produce the extraction results in the Resource Editor view (View > Resource Editor). These resources appear in the form of libraries and dictionaries in this view. You can customize the concepts and types directly within the libraries and dictionaries. See the topic [Working with Libraries](#) for more information.

- [Adding synonyms](#)
- [Adding concepts to types](#)
- [Excluding concepts from extraction](#)
- [Forcing Words into Extraction](#)

Adding synonyms

Synonyms associate two or more words that have the same meaning. Synonyms are often also used to group terms with their abbreviations or to group commonly misspelled words with the correct spelling. By using synonyms, the frequency for the target concept is greater, which makes it far easier to discover similar information that is presented in different ways in your text data.

The linguistic resource templates and libraries delivered with the product contain many predefined synonyms. However, if you discover unrecognized synonyms, you can define them so that they will be recognized the next time you extract.

The first step is to decide what the target, or lead, concept will be. The *target concept* is the word or phrase under which you want to group all synonym terms in the final results. During extraction, the synonyms are grouped under this target concept. The second step is to identify all of the synonyms for this concept. The target concept is substituted for all synonyms in the final extraction. A term must be extracted to be a synonym. However, the target concept does not need to be extracted for the substitution to occur. For example, if you want **intelligent** to be replaced by **smart**, then **intelligent** is the synonym and **smart** is the target concept.

If you create a new synonym definition, a new target concept is added to the dictionary. You must then add synonyms to that target concept. Whenever you create or edit synonyms, these changes are recorded in synonym dictionaries in the Resource Editor. If you want to view the entire contents of these synonym dictionaries or if you want to make a substantial number of changes, you may prefer to work directly in the Resource Editor. See the topic [Substitution/Synonym dictionaries](#) for more information.

Any new synonyms will automatically be stored in the first library listed in the library tree in the Resource Editor view—by default, this is the Local Library.

Note: If you look for a synonym definition and cannot find it through the context menus or directly in the Resource Editor, a match may have resulted from an internal fuzzy grouping technique. See the topic [Fuzzy Grouping](#) for more information.

To create a new synonym

1. In either the Extraction Results pane , Data pane, Category Definitions dialog box, or Cluster Definitions dialog box, select the concept(s) for which you want to create a new synonym.
2. From the menus, choose **Edit > Add to Synonym > New**. The Create Synonym dialog box opens.
3. Enter a target concept in the Target text box. This is the concept under which all of the synonyms will be grouped.
4. If you want to add more synonyms, enter them in the Synonyms list box. Use the global separator to separate each synonym term. See the topic [Options: Session Tab](#) for more information.
5. Click OK to apply your changes. The dialog box closes and the Extraction Results pane background color changes, indicating that you need to reextract to see your changes. If you have several changes, make them before you reextract.

To add a synonym

1. In either the Extraction Results pane , Data pane, Category Definitions dialog box, or Cluster Definitions dialog box, select the concept(s) that you want to add to an existing synonym definition.
2. From the menus, choose **Edit > Add to Synonym**. The menu displays a set of the synonyms with the most recently created at the top of the list. Select the name of the synonym to which you want to add the selected concept(s). If you see the synonym that you are looking for, select it, and the concept(s) selected are added to that synonym definition. If you do not see it, select **More** to display the All Synonyms dialog box.
3. In the All Synonyms dialog box, you can sort the list by natural sort order (order of creation) or in ascending or descending order. Select the name of the synonym to which you want to add the selected concept(s) and click OK. The dialog box closes, and the concepts are added to the synonym definition.

Adding concepts to types

Whenever an extraction is run, the extracted concepts are assigned to types in an effort to group terms that have something in common. IBM® SPSS® Modeler Text Analytics is delivered with many built-in types. See the topic [Built-in types](#) for more information. For most languages, concepts that are not found in any type dictionary but are extracted from the text are automatically typed as **<Unknown>**

When reviewing your results, you may find some concepts that appear in one type that you want assigned to another, or you may find that a group of words really belongs in a new type by itself. In these cases, you would want to reassign the concepts to another type or create a new type altogether.

For example, suppose that you are working with survey data relating to automobiles and you are interested in categorizing by focusing on different areas of the vehicles. You could create a type called **<Dashboard>** to group all of the concepts relating to gauges and knobs found on the dashboard of the vehicles. Then you could assign concepts such as **gas gauge**, **heater**, **radio**, and **odometer** to that new type.

In another example, suppose that you are working with survey data relating to universities and colleges and the extraction typed **Johns Hopkins** (the university) as a <Person> type rather than as an <Organization> type. In this case, you could add this concept to the <Organization> type.

Whenever you create a type or add concepts to a type's term list, these changes are recorded in type dictionaries within your linguistic resource libraries in the Resource Editor. If you want to view the contents of these libraries or make a substantial number of changes, you may prefer to work directly in the Resource Editor. See the topic [Adding terms](#) for more information.

To add a concept to a type

1. In either the Extraction Results pane , Data pane, Category Definitions dialog box, or Cluster Definitions dialog box, select the concept(s) that you want to add to an existing type.
2. Right-click to open the context menu.
3. From the menus, choose **Edit** > **Add to Type**. The menu displays a set of the types with the most recently created at the top of the list. Select the type name to which you want to add the selected concept(s). If you see the type name that you are looking for, select it, and the concept(s) selected are added to that type. If you do not see it, select **More** to display the All Types dialog box.
4. In the All Types dialog box, you can sort the list by natural sort (order of creation) or in ascending or descending order. Select the name of the type to which you want to add the selected concept(s) and click **OK**. The dialog box closes, and they are added as terms to the type.

To create a new type

1. In either the Extraction Results pane , Data pane, Category Definitions dialog box, or Cluster Definitions dialog box, select the concepts for which you want to create a new type.
2. From the menus, choose **Edit** > **Add to Type** > **New**. The Type Properties dialog box opens.
3. Enter a new name for this type in the Name text box and make any changes to the other fields. See the topic [Creating types](#) for more information.
4. Click **OK** to apply your changes. The dialog box closes and the Extraction Results pane background color changes, indicating that you need to re-extract to see your changes. If you have several changes, make them before you re-extract.

Excluding concepts from extraction

When reviewing your results, you may occasionally find concepts that you did not want extracted or used by any automated category building techniques. In some cases, these concepts have a very high frequency count and are completely insignificant to your analysis. In this case, you can mark a concept to be excluded from the final extraction. Typically, the concepts you add to this list are fill-in words or phrases used in the text for continuity but that do not add anything important and may clutter the extraction results. By adding concepts to the exclude dictionary, you can make sure that they are never extracted.

By excluding concepts, all variations of the excluded concept disappear from your extraction results the next time that you extract. If this concept already appears as a descriptor in a category, it will remain in the category with a zero count after re-extraction.

When you exclude, these changes are recorded in an exclude dictionary in the Resource Editor. If you want to view all of the exclude definitions and edit them directly, you may prefer to work directly in the Resource Editor. See the topic [Exclude dictionaries](#) for more information.

To exclude concepts

1. In either the Extraction Results pane , Data pane, Category Definitions dialog box, or Cluster Definitions dialog box, select the concept(s) that you want to exclude from the extraction.
2. Right-click to open the context menu.
3. Select **Exclude from Extraction**. The concept is added to the exclude dictionary in the Resource Editor and the Extraction Results pane background color changes, indicating that you need to re-extract to see your changes. If you have several changes, make them before you re-extract.

Note: Any words that you exclude will automatically be stored in the first library listed in the library tree in the Resource Editor—by default, this is the Local Library.

Forcing Words into Extraction

When reviewing the text data in the Data pane after extraction, you may discover that some words or phrases were not extracted. Often, these words are verbs or adjectives that you are not interested in. However, sometimes you do want to use a word or phrase that was not extracted as part of a category definition.

If you would like to have these words and phrases extracted, you can force a term into a type library. See the topic [Forcing terms](#) for more information.

Important! Marking a term in a dictionary as forced is not foolproof. By this, we mean that even though you have explicitly added a term to a dictionary, there are times when it may not be present in the Extraction Results pane after you have reextracted or it does appear but not exactly as you have declared it. Although this occurrence is rare, it can happen when a word or phrase was already extracted as part of a longer phrase. To prevent this, apply the Entire (no compounds) match option to this term in the type dictionary. See the topic [Adding terms](#) for more information.

Related information

- [Refining extraction results](#)
 - [Adding synonyms](#)
 - [Adding concepts to types](#)
 - [Excluding concepts from extraction](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Categorizing text data

In the Categories and Concepts view , you can create *categories* that represent, in essence, higher-level concepts, or topics, that will capture the key ideas, knowledge, and attitudes expressed in the text.

As of the release of IBM® SPSS® Modeler Text Analytics 14, categories can also have a hierarchical structure, meaning they can contain subcategories and those subcategories can also have subcategories of their own and so on. You can import predefined category structures, formerly called code frames, with hierarchical categories as well as build these hierarchical categories inside the product.

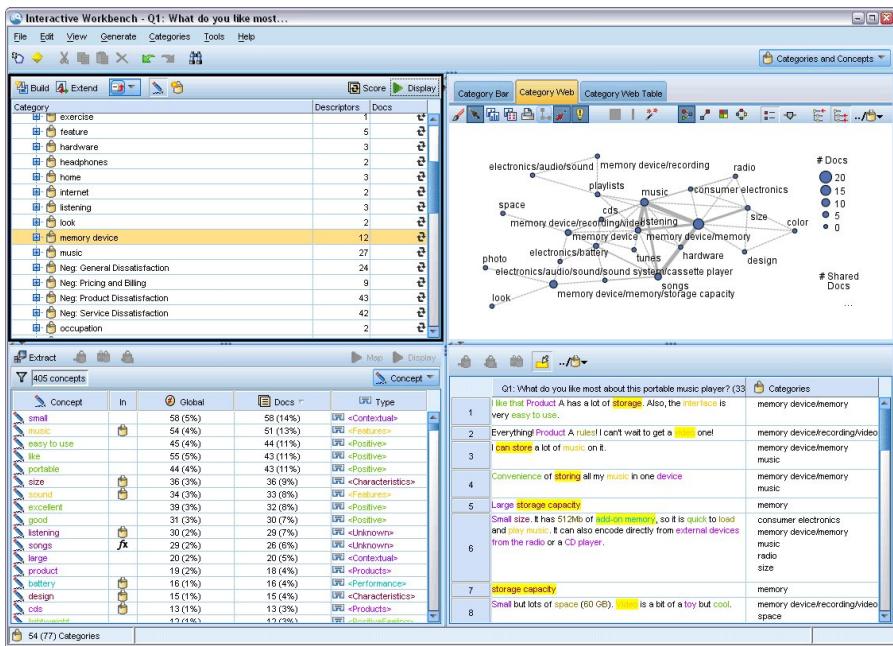
In effect, hierarchical categories enable you to build a tree structure with one or more subcategories to group items such as different concept or topic areas more accurately. A simple example can be related to leisure activities; answering a question such as *What activity would you like to do if you had more time?* you may have top categories such as *sports, art and craft, fishing*, and so on; down a level, below *sports*, you may have subcategories to see if this is *ball games, water-related*, and so on.

Categories are made up of a set of descriptors, such as *concepts, types, patterns* and *category rules*. Together, these descriptors are used to identify whether or not a document or record belongs to a given category. The text within a document or record can be scanned to see whether any text matches a descriptor. If a match is found, the document/record is assigned to that category. This process is called *categorization*.

You can work with, build, and visually explore your categories using the data presented in the four panes of the Categories and Concepts view, each of which can be hidden or shown by selecting its name from the View menu.

- Categories pane. Build and manage your categories in this pane. See the topic [The Categories Pane](#) for more information.
- Extraction Results pane. Explore and work with the extracted concepts and types in this pane. See the topic [Extraction results: Concepts and types](#) for more information.
- Visualization pane. Visually explore your categories and how they interact in this pane. See the topic [Category Graphs and Charts](#) for more information.
- Data pane. Explore and review the text contained within documents and records that correspond to selections in this pane. See the topic [The Data Pane](#) for more information.

Figure 1. Categories and Concepts view



While you might start with a set of categories from a text analysis package (TAP) or import from a predefined category file, you might also need to create your own. Categories can be created automatically using the product's robust set of automated techniques, which use extraction results (concepts, types, and patterns) to generate categories and their descriptors. Categories can also be created manually using additional insight you may have regarding the data. However, you can only create categories manually or fine-tune them through the interactive workbench. See the topic [Text Mining Node: Model Tab](#) for more information. You can create category definitions manually by dragging and dropping extraction results into the categories. You can enrich these categories or any empty category by adding category rules to a category, using your own predefined categories, or a combination.

Each of the techniques and methods is well suited for certain types of data and situations, but often it will be helpful to combine techniques in the same analysis to capture the full range of documents or records. And in the course of categorization, you may see other changes to make to the linguistic resources.

- [The Categories Pane](#)
- [Methods and Strategies for Creating Categories](#)
- [About Categories](#)
- [The Data Pane](#)
- [Building categories](#)
- [Extending categories](#)
- [Creating Categories Manually](#)
- [Using Category Rules](#)
- [Importing and Exporting Predefined Categories](#)
- [Using Text Analysis Packages](#)
- [Editing and Refining Categories](#)

The Categories Pane

The Categories pane is the area in which you can build and manage your categories. This pane is located in the upper left corner of the Categories and Concepts view. After extracting the concepts and types from your text data, you can begin building categories automatically using techniques such as concept inclusion, co-occurrence, and so on or manually. See the topic [Building categories](#) for more information.

Each time a category is created or updated, the documents or records can be scored by clicking the Score button to see whether any text matches a descriptor in a given category. If a match is found, the document or record is assigned to that category. The end result is that most, if not all, of the documents or records are assigned to categories based on the descriptors in the categories.

Note: If there are more categories that can fit in the visible pane, you can use the controls at the bottom of the pane to move forwards and backwards through the categories, or enter a page number to go to.

Category Tree Table

The tree table in this pane presents the set of categories, subcategories, and descriptors. The tree also has several columns presenting information for each tree item. The following columns may be available for display:

- **Code** Lists the code value for each category. This column is hidden by default. You can display this column through the menus: View > Categories Pane.
- **Category**. Contains the category tree showing the name of the category and subcategories. Additionally if the descriptors toolbar icon is clicked, the set of descriptors will also be displayed.
- **Descriptors**. Provides the number of descriptors that make up its definition. This count does not include the number of descriptors in the subcategories. No count is given when a descriptor name is shown in the Categories column. You can display or hide the descriptors themselves in the tree through the menus: View > Categories Pane > All Descriptors.
- **Docs** After scoring, this column provides the number of documents or records that are categorized into a category and all of its subcategories. So if 5 records match your top category based on its descriptors, and 7 different records match a subcategory based on its descriptors, the total doc count for the top category is a sum of the two-- in this case it would be 12. However, if the same record matched the top category and its subcategory, then the count would be 11.

When no categories exist, the table still contains two rows. The top row, called All Documents, is the total number of documents or records. A second row, called Uncategorized, shows the number of documents/records that have yet to be categorized.

For each category in the pane, a small yellow bucket icon precedes the category name. If you double-click a category, or choose View > Category Definitions in the menus, the Category Definitions dialog box opens and presents all of the elements, called *descriptors*, that make up its definition, such as concepts, types, patterns, and category rules. See the topic [About Categories](#) for more information. By default, the category tree table does not show the descriptors in the categories. If you want to see the descriptors directly in the tree rather than in the Category Definitions dialog box, click the toggle button with the pencil icon in the toolbar. When this toggle button is selected, you can expand your tree to see the descriptors as well.

Scoring Categories

The Docs. column in the category tree table displays the number of documents or records that are categorized into that specific category. If the numbers are out of date or are not calculated, an icon appears in that column. You can click Score on the pane toolbar to recalculate the number of documents. Keep in mind that the scoring process can take some time when you are working with larger datasets.

Selecting Categories in the Tree

When making selections in the tree, you can only select sibling categories -- that is to say, if you select top level categories, you can not also select a subcategory. Or if you select 2 subcategories of a given category, you cannot simultaneously select a subcategory of another category. Selecting a discontiguous category will result in the loss of the previous selection.

Displaying in Data and Visualization Panes

When you select a row in the table, you can click the Display button to refresh the Visualization and Data panes with information corresponding to your selection. If a pane is not visible, clicking Display will cause the pane to appear.

Refining Your Categories

Categorization may not yield perfect results for your data on the first try, and there may well be categories that you want to delete or combine with other categories. You may also find, through a review of the extraction results, that there are some categories that were not created that you would find useful. If so, you can make manual changes to the results to fine-tune them for your particular context. See the topic [Editing and Refining Categories](#) for more information.

Related information

- [Categorizing text data](#)
- [About Categories](#)
- [The Data Pane](#)
- [Building categories](#)
- [About linguistic techniques](#)
- [Extending categories](#)
- [Creating Categories Manually](#)
- [Using Category Rules](#)
- [Using Text Analysis Packages](#)
- [Editing and Refining Categories](#)
- [Microsoft Internet Explorer settings for Help](#)

Methods and Strategies for Creating Categories

If you have not yet extracted or your extraction results are out of date, the use of one of the category building or extending techniques will prompt you for an extraction automatically. After you have applied a technique, the concepts and types that were grouped into a category are still available for category building with other techniques. This means that you may see a concept in multiple categories unless you choose not to reuse them.

In order to help you create the best categories, please review the following:

- **Methods for creating categories**. See the topic [Methods for Creating Categories](#) for more information.
 - **Strategies for creating categories**. See the topic [Strategies for Creating Categories](#) for more information.
 - **Tips for creating categories**. See the topic [Tips for Creating Categories](#) for more information.
-
- [Methods for Creating Categories](#)
 - [Strategies for Creating Categories](#)
 - [Tips for Creating Categories](#)
 - [Choosing the Best Descriptors](#)

Related information

- [Methods for Creating Categories](#)
- [Strategies for Creating Categories](#)
- [Tips for Creating Categories](#)
- [Choosing the Best Descriptors](#)
- [About Categories](#)
- [Category Properties](#)
- [Category Relevance](#)
- [Microsoft Internet Explorer settings for Help](#)

Methods for Creating Categories

Because every dataset is unique, the number of category creation methods and the order in which you apply them may change over time. Additionally, since your text mining goals may be different from one set of data to the next, you may need to experiment with the different methods to see which one produces the best results for the given text data. None of the automatic techniques will perfectly categorize your data; therefore we recommend finding and applying one or more automatic techniques that work well with your data.

Besides using text analysis packages (TAPs, *.tap) with prebuilt category sets, you can also categorize your responses using any combination of the following methods:

- **Automatic building techniques.** Several linguistic-based and frequency-based category options are available to automatically build categories for you. See the topic [Building categories](#) for more information.
- **Automatic extending techniques.** Several linguistic techniques are available to extend existing categories by adding and enhancing descriptors so that they capture more records. See the topic [Extending categories](#) for more information.
- **Manual techniques.** There are several manual methods, such as drag-and-drop. See the topic [Creating Categories Manually](#) for more information.

Related information

- [Methods and Strategies for Creating Categories](#)
- [Strategies for Creating Categories](#)
- [Tips for Creating Categories](#)
- [Choosing the Best Descriptors](#)
- [About Categories](#)
- [Category Properties](#)
- [Category Relevance](#)
- [Microsoft Internet Explorer settings for Help](#)

Strategies for Creating Categories

The following list of strategies is by no means exhaustive but it can provide you with some ideas on how to approach the building of your categories.

- When you define the Text Mining node, select a category set from a text analysis package (TAP) so that you begin your analysis with some prebuilt categories. These categories may sufficiently categorize your text right from the start. However, if you want to add more

categories, you can edit the Build Categories settings (Categories > Build Settings). Open the Advanced Settings: Linguistics dialog and choose the Category input option Unused extraction results and build the additional categories.

- When you define the node, select a category set from a TAPin the Categories and Concepts view in the Interactive Workbench. Next, drag and drop unused concepts or patterns into the categories as you deem appropriate. Then, extend the existing categories you've just edited (Categories > Extend Categories) to obtain more descriptors that are related to the existing category descriptors.
- Build categories automatically using the advanced linguistic settings (Categories > Build Categories). Then, refine the categories manually by deleting descriptors, deleting categories, or merging similar categories until you are satisfied with the resulting categories. Additionally, if you originally built categories **without** using the Generalize with wildcards where possible option, you can also try to simplify the categories automatically using the Extend Categories using the Generalize option.
- Import a predefined category file with very descriptive category names and/or annotations. Additionally, if you originally imported **without** choosing the option to import or generate descriptors from category names, you can later use the Extend Categories dialog and choose the Extend empty categories with descriptors generated from the category name. option. Then, extend those categories a second time but use the grouping techniques this time.
- Manually create a first set of categories by sorting concepts or concept patterns by frequency and then dragging and dropping the most interesting ones to the Categories pane. Once you have that initial set of categories, use the Extend feature (Categories > Extend Categories) to expand and refine all of the selected categories so they'll include other related descriptors and thereby match more records.

After applying these techniques, we recommend that you review the resulting categories and use manual techniques to make minor adjustments, remove any misclassifications, or add records or words that may have been missed. Additionally, since using different techniques may produce redundant categories, you could also merge or delete categories as needed. See the topic [Editing and Refining Categories](#) for more information.

Related information

- [Methods and Strategies for Creating Categories](#)
- [Methods for Creating Categories](#)
- [Tips for Creating Categories](#)
- [Choosing the Best Descriptors](#)
- [About Categories](#)
- [Category Properties](#)
- [Category Relevance](#)
- [Microsoft Internet Explorer settings for Help](#)

Tips for Creating Categories

In order to help you create better categories, you can review some tips that can help you make decisions on your approach.

Tips on Category-to-Document Ratio

The categories into which the documents and records are assigned are not often mutually exclusive in qualitative text analysis for at least two reasons:

- First, a general rule of thumb says that the longer the text document or record, the more distinct the ideas and opinions expressed. Thus, the chances that a document or record can be assigned multiple categories is greatly increased.
- Second, often there are various ways to group and interpret text documents or records that are not logically separate. In the case of a survey with an open-ended question about the respondent's political beliefs, we could create categories, such as *Liberal* and *Conservative*, or *Republican* and *Democrat*, as well as more specific categories, such as *Socially Liberal*, *Fiscally Conservative*, and so forth. These categories do not have to be mutually exclusive and exhaustive.

Tips on Number of Categories to Create

Category creation should flow directly from the data—as you see something interesting with respect to your data, you can create a category to represent that information. In general, there is no recommended upper limit on the number of categories that you create. However, it is certainly possible to create too many categories to be manageable. Two principles apply:

- **Category frequency.** For a category to be useful, it has to contain a minimum number of documents or records. One or two documents may include something quite intriguing, but if they are one or two out of 1,000 documents , the information they contain may not be frequent enough in the population to be practically useful.
- **Complexity.** The more categories you create, the more information you have to review and summarize after completing the analysis. However, too many categories, while adding complexity, may not add useful detail.

Unfortunately, there are no rules for determining how many categories are too many or for determining the minimum number of records per category. You will have to make such determinations based on the demands of your particular situation.

We can, however, offer advice about where to start. Although the number of categories should not be excessive, in the early stages of the analysis it is better to have too many rather than too few categories. It is easier to group categories that are relatively similar than to split off cases into

new categories, so a strategy of working from more to fewer categories is usually the best practice. Given the iterative nature of text mining and the ease with which it can be accomplished with this software program, building more categories is acceptable at the start.

Related information

- [Methods and Strategies for Creating Categories](#)
 - [Methods for Creating Categories](#)
 - [Strategies for Creating Categories](#)
 - [Choosing the Best Descriptors](#)
 - [About Categories](#)
 - [Category Properties](#)
 - [Category Relevance](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Choosing the Best Descriptors

The following information contains some guidelines for choosing or making the best descriptors (concepts, types, TLA patterns, and category rules) for your categories. Descriptors are the building blocks of categories. When some or all of the text in a document or record matches a descriptor, the document or record is matched to the category.

Unless a descriptor contains or corresponds to an extracted concept or pattern, it will not be matched to any documents or records. Therefore, use concepts, types, patterns, and category rules as described in the following paragraphs.

Since concepts represent not only themselves but also a set of underlying terms that can range from plural/singular forms, to synonyms, to spelling variations, only the concept itself should be used as a descriptor or as part of a descriptor. To learn more about the underlying terms for any given concept, click on the concept name in the Extraction Results pane of the Categories and Concepts view. When you hover over the concept name, a tooltip appears and displays any of the underlying terms found in your text during the last extraction. Not all concepts have underlying terms. For example, if **car** and **vehicle** were synonyms but **car** was extracted as the concept with **vehicle** as an underlying term, then you only want to use **car** in a descriptor since it will automatically match document or records with **vehicle**.

Concepts and Types as Descriptors

Use a concept as a descriptor when you want to find all documents or records containing that concept (or any of its underlying terms). In this case, the use of a more complex category rule is not needed since the exact concept name is sufficient. Keep in mind that when you use resources that extract opinions, sometimes concepts can change during TLA pattern extraction to capture the truer sense of the sentence (refer to the example in the next section on TLA).

For example, a survey response indicating each person's favorite fruits such as "*Apple and pineapple are the best*" could result in the extraction of **apple** and **pineapple**. By adding the concept **apple** as a descriptor to your category, all responses containing the concept **apple** (or any of its underlying terms) are matched to that category.

However, if you are interested in simply knowing which responses mention **apple** in any way, you can write a category rule such as * **apple** * and you will also capture responses that contain concepts such as **apple**, **apple sauce**, or **french apple tart**.

You can also capture all the documents or records that contain concepts that were typed the same way by using a type as a descriptor directly such as <Fruit>. Please note that you cannot use * with types.

See the topic [Extraction results: Concepts and types](#) for more information.

Text Link Analysis (TLA) Patterns as Descriptors

Use a TLA pattern result as a descriptor when you want to capture finer, nuanced ideas. When text is analyzed during TLA extraction, the text is processed one sentence, or clause, at a time rather than looking at the entire text (the document or record). By considering all of the parts of a single sentence together, TLA can identify opinions, relationships between two elements, or a negation, for example, and understand the truer sense. You can use concept patterns or type patterns as descriptors. See the topic [Type and Concept Patterns](#) for more information.

For example, if we had the text "*the room was not that clean*", the following concepts could be extracted: **room** and **clean**. However, if TLA extraction was enabled in the extraction setting, TLA could detect that **clean** was used in a negative way and actually corresponds to **not clean**, which is a synonym of the concept **dirty**. Here, you can see that using the concept **clean** as a descriptor on its own would match this text but could also capture other document or records mentioning cleanliness. Therefore, it might be better to use the TLA concept pattern with **dirty** as output concept since it would match this text and likely be a more appropriate descriptor.

Category Business Rules as Descriptors

Category rules are statements that automatically classify documents or records into a category based on a logical expression using extracted concepts, types, and patterns as well as Boolean operators. For example, you could write an expression that means *include all records that contain the extracted concept **embassy** but not **argentina** in this category*.

You can write and use category rules as descriptors in your categories to express several different ideas using &, |, and ! () Booleans. For detailed information on the syntax of these rules and how to write and edit them, see [Using Category Rules](#).

- Use a category rule with the & (AND) Boolean operator to help you find documents or records in which 2 or more concepts occur. The 2 or more concepts connected by & operators do not need to occur in the same sentence or phrase, but can occur anywhere in the same document or record to be considered a match to the category. For example, if you create the category rule `food & cheap` as a descriptor, it would match a record containing the text, "the food was pretty expensive, but the rooms were cheap" despite the fact that `food` was not the noun being called `cheap` since the text contained both `food` and `cheap`.
- Use a category rule with the ! () (NOT) Boolean operator as a descriptor to help you find documents or records in which some things occur but others do not. This can help avoid grouping information that may seem related based on words but not on context. For example, if you create the category rule `<Organization> & !(ibm)` as a descriptor, it would match the following text `SPSS Inc. was a company founded in 1967` and not match the following text `the software company was acquired by IBM..`
- Use a category rule with the | (OR) Boolean operator as a descriptor to help you find documents or records containing one of several concepts or types. For example, if you create the category rule `(personnel|staff|team|coworkers) & bad` as a descriptor, it would match any documents or records in which any of those nouns are found with the concept `bad`.
- Use types in category rules to make them more generic and possibly more deployable. For example, if you were working with hotel data, you might be very interested in learning what customers think about hotel personnel. Related terms might include words such as receptionist, waiter, waitress, reception desk, front desk and so on. You could, in this case, create a new type called `<HotelStaff>` and add all of the preceding terms to that type. While it is possible to create one category rule for every kind of staff such as `[* waitress * & nice], [* desk * & friendly], [* receptionist * & accommodating]`, you could create a single, more generic category rule using the `<HotelStaff>` type to capture all responses that have favorable opinions of the hotel staff in the form of `[<HotelStaff> & <Positive>]`.

Note: You can use both + and & in category rules when including TLA patterns in those rules. See the topic [Using TLA Patterns in Category Rules](#) for more information.

Example of how concepts, TLA, or category rules as descriptors match differently

The following example demonstrates how using a concept as a descriptor, category rule as a descriptor, or using a TLA pattern as a descriptor affects how documents or records are categorized. Let's say you had the following 5 records.

- A: "awesome restaurant staff, excellent food and rooms comfortable and clean."
- B: "restaurant personnel was awful, but rooms were clean."
- C: "Comfortable, clean rooms."
- D: "My room was not that clean."
- E: "Clean."

Since the records include the word `clean` and you want to capture this information, you could create one of the descriptors shown in the following table. Based on the essence you are trying to capture, you can see how using one kind of descriptor over another can produce different results.

Table 1. How Example Records Matched Descriptors

Descriptor	A	B	C	D	E	Explanation
<code>clean</code>	match	match	match	match	match	Descriptor is an extracted concept. Every record contained the concept <code>clean</code> , even record D since without TLA, it is not known automatically that "not <code>clean</code> " means <code>dirty</code> by the TLA rules.
<code>clean + .</code>	-	-	-	-	match	Descriptor is a TLA pattern that represents <code>clean</code> by itself. Matched only the record where <code>clean</code> was extracted with no associated concept during TLA extraction.
<code>[clean]</code>	match	match	match	-	match	Descriptor is a category rule that looks for a TLA rule that contains <code>clean</code> on its own or with something else. Matched all records where a TLA output containing <code>clean</code> was found regardless of whether <code>clean</code> was linked to another concept such as <code>room</code> and in any slot position.

Related information

- [Methods and Strategies for Creating Categories](#)
 - [Methods for Creating Categories](#)
 - [Strategies for Creating Categories](#)
 - [Tips for Creating Categories](#)
 - [About Categories](#)
 - [Category Properties](#)
 - [Category Relevance](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

About Categories

Categories refer to a group of closely related concepts, opinions, or attitudes. To be useful, a category should also be easily described by a short phrase or label that captures its essential meaning.

For example, if you are analyzing survey responses from consumers about a new laundry soap, you can create a category labeled *odor* that contains all of the responses describing the smell of the product. However, such a category would not differentiate between those who found the smell pleasant and those who found it offensive. Since IBM® SPSS® Modeler Text Analytics is capable of extracting opinions when using the appropriate resources, you could then create two other categories to identify respondents who *enjoyed the odor* and respondents who *disliked the odor*.

You can create and work with your categories in the Categories pane in the upper left pane of the Categories and Concepts view window. Each category is defined by one or more descriptors. **Descriptors** are concepts, types, and patterns, as well as category rules that have been used to define a category.

If you want to see the descriptors that make up a given category, you can click the pencil icon in the Categories pane toolbar and then expand the tree to see the descriptors. Alternatively, select the category and open the Category Definitions dialog box (View > Category Definitions).

When you build categories automatically using category building techniques such as concept inclusion, the techniques will use concepts and types as the descriptors to create your categories. If you extract TLA patterns, you can also add patterns or parts of those patterns as category descriptors. See the topic [Exploring Text Link Analysis](#) for more information. And if you build clusters, you can add the concepts in a cluster to new or existing categories. Lastly, you can manually create category rules to use as descriptors in your categories. See the topic [Using Category Rules](#) for more information.

- [Category Properties](#)

Related information

- [Categorizing text data](#)
 - [The Categories Pane](#)
 - [The Data Pane](#)
 - [Building categories](#)
 - [About linguistic techniques](#)
 - [Extending categories](#)
 - [Creating Categories Manually](#)
 - [Using Category Rules](#)
 - [Using Text Analysis Packages](#)
 - [Editing and Refining Categories](#)
 - [Methods and Strategies for Creating Categories](#)
 - [Methods for Creating Categories](#)
 - [Strategies for Creating Categories](#)
 - [Tips for Creating Categories](#)
 - [Choosing the Best Descriptors](#)
 - [Category Properties](#)
 - [Category Relevance](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Category Properties

In addition to descriptors, categories also have properties you can edit in order to rename categories, add a label, or add an annotation.

The following properties exist:

- **Name.** This name appears in the tree by default. When a category is created using an automated technique, it is given a name automatically.

- **Label.** Using labels is helpful in creating more meaningful category descriptions for use in other products or in other tables or graphs. If you choose the option to display the label, then the label is used in the interface to identify the category.
- **Code.** The code number corresponds to the code value for this category. .
- **Annotation.** You can add a short description for each category in this field. When a category is generated by the Build Categories dialog, a note is added to this annotation automatically. You can also add sample text to an annotation directly from the Data pane by selecting the text and choosing Categories > Add to Annotation from the menus.

Related information

- [Methods and Strategies for Creating Categories](#)
 - [Methods for Creating Categories](#)
 - [Strategies for Creating Categories](#)
 - [Tips for Creating Categories](#)
 - [Choosing the Best Descriptors](#)
 - [About Categories](#)
 - [Category Relevance](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

The Data Pane

As you create categories, there may be times when you might want to review some of the text data you are working with. For example, if you create a category in which 640 documents are categorized, you might want to look at some or all of those documents to see what text was actually written. You can review records or documents in the Data pane, which is located in the lower right. If not visible by default, choose View > Panes > Data from the menus.

The Data pane presents one row per document or record corresponding to the selection in the Categories pane, Extraction Results pane, or the Category Definitions dialog box up to a certain display limit. By default, the number of documents or records shown in the Data pane is limited in order to allow you to see your data more quickly. However, you can adjust this in the Options dialog box. If you are dealing with very large datasets, the speed of display may be improved by turning off the option to show categories. See the topic [Options: Session Tab](#) for more information.

Note: If there are more records that can fit in the visible pane, you can use the controls at the bottom of the pane to move forwards and backwards through the records, or enter a page number to go to.

Displaying and refreshing the Data pane

The Data pane does not refresh its display automatically because with larger datasets automatic data refreshing could take some time to complete. Therefore, whenever you make a selection in another pane in this view or the Category Definitions dialog box, click Display to refresh the contents of the Data pane.

Text Documents or Records

If your text data is in the form of records and the text is relatively short in length, the text field in the Data pane displays the text data in its entirety. However, when working with records and larger datasets, the text field column shows a short piece of the text and opens a Text Preview pane to the right to display more or all of the text of the record you have selected in the table. If your text data is in the form of individual documents, the Data pane shows the document's filename. When you select a document, the Text Preview pane opens with the selected document's text.

Colors and Highlighting

Whenever you display the data, concepts and descriptors found in those documents or records are highlighted in color to help you easily identify them in the text. The color coding corresponds to the types to which the concepts belong. You can also hover your mouse over color-coded items to display the concept under which it was extracted and the type to which it was assigned. Any text that was not extracted appears in black. Typically, these unextracted words are often connectors (*and* or *with*), pronouns (*me* or *they*), and verbs (*is*, *have*, or *take*).

Data Pane Columns

While the text field column is always visible, you can also display other columns. To display other columns, choose View > Data Pane from the menus, and then select the column that you want to display in the Data pane. The following columns may be available for display:

- "Text field name" (#)/Documents. Adds a column for the text data from which concepts and type were extracted. If your data is in documents, the column is called Documents and only the document filename or full path is visible. To see the text for those documents you must look in the Text Preview pane. The number of rows in the Data pane is shown in parentheses after this column name. There may be times when not all documents or records are shown due to a limit in the Options dialog used to increase the speed of loading. If the maximum is reached, the number will be followed by - Max. See [Options: Session Tab](#) for more information.
- Categories. Lists each of the categories to which a record belongs. Whenever this column is shown, refreshing the Data pane may take a bit longer so as to show the most up-to-date information.

- Force In. Lists the categories into which you have forced a document. Documents can be forced into the category through the Edit > Force In menu selection. See [Forcing documents into categories](#) for more information.
 - Force Out. Lists the categories from which you have removed a document. Documents can be forced out of a category through the Edit > Force Out menu selection. For example, this might be used when a respondent's sarcasm causes a response to be miscategorized. See [Forcing documents into categories](#) for more information.
 - Category Counts. Lists the number of the categories to which a record belongs.
 - Relevance Ranks. Provides a rank for each record in a single category. This rank shows how well the record fits into the category compared to the other records in that category. Select a category in the Categories pane (upper left pane) to see the rank. See [Category Relevance](#) for more information.
 - Response Flags. Adds a column that shows any flags you may be using. Click inside this column to change the type of flag that you assign to documents. You might flag documents with a "complete" flag or an "important" flag, or remove flags. This is useful for reviewing the completeness of a category model. See [Flagging responses](#) for more information.
- [Category Relevance](#)
 - [Flagging responses](#)
-

Category Relevance

To help you build better categories, you can review the relevance of the documents or records in each category as well as the relevance of all categories to which a document or record belongs.

Relevance of a Category to a Record

Whenever a document or record appears in the Data pane, all categories to which it belongs are listed in the Categories column. When a document or record belongs to multiple categories, the categories in this column appear in order from the most to the least relevant match. The category listed first is thought to correspond best to this document or record. See the topic [The Data Pane](#) for more information.

Relevance of a Record to a Category

When you select a category, you can review the relevance of each of its records in the Relevance Rank column in the Data pane. This relevance rank indicates how well the document or record fits into the selected category compared to the other records in that category. To see the rank of the records for a single category, select this category in the Categories pane (upper left pane) and the rank for document or record appears in the column. This column is not visible by default but you can choose to display it. See the topic [The Data Pane](#) for more information.

The lower the number for the record's rank, the better the fit or the more relevant this record is to the selected category such that 1 is the best fit. If more than one record has the same relevance, each appears with the same rank followed by an equal sign (=) to denote they have equal relevance. For example, you might have the following ranks 1=, 1=, 3, 4, and so on, which means that there are two records that are equally considered as best matches for this category.

Tip: You could add the text of the most relevant record to the category annotation to help provide a better description of the category. Add the text directly from the Data pane by selecting the text and choosing Categories > Add to Annotation from the menus.

Related information

- [Methods and Strategies for Creating Categories](#)
 - [Methods for Creating Categories](#)
 - [Strategies for Creating Categories](#)
 - [Tips for Creating Categories](#)
 - [Choosing the Best Descriptors](#)
 - [About Categories](#)
 - [Category Properties](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Flagging responses

To help you monitor your progress, you can mark documents using flags in the Data pane. This feature is only available if the source document contains a unique ID. If the source document doesn't contain a unique ID, you can add a Derive node between the source document and the Text Mining node.

There are many reasons why you might want to mark a document, including:

- To mark off documents that you have manually reviewed so that you know where to pick up later
- To mark off a document that you are unsure about how to handle

Once you mark a document with a flag, you can continue to work with the documents. They are purely for your own record-keeping. You can choose from the following flags:

Table 1. Flag descriptions

Flag	Description
	Complete flag to denote documents that you deem finished.
	Important flag to denote documents that you deem important.

To mark a document with a flag:

1. From within the Data pane, right-click the document that you want to mark.
2. From the context menu, choose View > Data Pane > Response Flags and then select the type of flag that you want to use (Important Flag or Complete Flag). The selected flag is assigned. If the Flag column in the Data pane is not visible, it appears.

To clear flags:

1. From within the Data pane, right-click the documents for which you want to remove a flag.
2. From the context menu, choose Mark Responses With > Clear Flags. The selected flags are removed.

Building categories

While you may have categories from a text analysis package, you can also build categories automatically using a number of linguistic and frequency techniques. Through the Build Categories Settings dialog box, you can apply the automated linguistic and frequency techniques to produce categories from either concepts or from concept patterns.

In general, categories can be made up of different kinds of descriptors (types, concepts, TLA patterns, category rules). When you build categories using the automated category building techniques, the resulting categories are named after a concept or concept pattern (depending on the input you select) and each contains a set of descriptors. These descriptors may be in the form of category rules or concepts and include all the related concepts discovered by the techniques.

After building categories, you can learn a lot about the categories by reviewing them in the Categories pane or exploring them through the graphs and charts. You can then use manual techniques to make minor adjustments, remove any misclassifications, or add records or words that may have been missed. After you have applied a technique, the concepts, types, and patterns that were grouped into a category are still available for other techniques. Also, since using different techniques may also produce redundant or inappropriate categories, you can also merge or delete categories. See the topic [Editing and Refining Categories](#) for more information.

Important! In earlier releases, co-occurrence and synonym rules were surrounded by square brackets. In this release, square brackets now indicate a text link analysis pattern result. Instead, co-occurrence and synonym rules will be encapsulated by parentheses such as **(speaker systems | speakers)**.

To build categories

1. From the menus, choose Categories > Build Categories. Unless you have chosen to never prompt, a message box is displayed.
2. Choose whether you want to build now or edit the settings first.
 - Click Build Now to begin building categories using the current settings. The settings selected by default are often sufficient to begin the categorization process. The category building process begins and a progress dialog appears.
 - Click Edit to review and modify the build settings.

Note: The maximum number of categories that can be displayed is 10,000. A warning is displayed if this number is reached or exceeded. If this happens you should change your Build or Extend Categories options to reduce the number of categories built.

Inputs

The categories are built from descriptors derived from either type patterns or types. In the table, you can select the individual types or patterns to include in the category building process.

Type patterns. If you select type patterns, categories are built from patterns rather than types and concepts on their own. In that way, any records or documents containing a concept pattern belonging to the selected type pattern are categorized. So, if you select the **<Budget>** and **<Positive>** type pattern in the table, categories such as **cost & <Positive>** or **rates & excellent** could be produced.

When using type patterns as input for automated category building, there are times when the techniques identify multiple ways to form the category structure. Technically, there is no single right way to produce the categories; however you might find one structure more suited to your analysis than another. To help customize the output in this case, you can designate a type as the preferred focus. All the top-level categories

produced will come from a concept of the type you select here (and no other type). Every subcategory will contain a text link pattern from this type. Choose this type in the Structure categories by pattern type: field and the table will be updated to show only the applicable patterns containing the selected type. More often than not, <Unknown> will be preselected for you. This results in all of the patterns containing the type <Unknown> being selected. The table displays the types in descending order starting with the one with the greatest number of records or documents (Doc. count).

Types. If you select types, the categories will be built from the concepts belonging to the selected types. So if you select the <Budget> type in the table, categories such as **cost** or **price** could be produced since **cost** and **price** are concepts assigned to the <Budget> type.

By default, only the types that capture the most records or documents are selected. This pre-selection allows you to quickly focus in on the most interesting types and avoid building uninteresting categories. The table displays the types in descending order starting with the one with the greatest number of records or documents (Doc. count). Types from the **Opinions** library are deselected by default in the types table.

The input you choose affects the categories you obtain. When you choose to use Types as input, you can see the clearly related concepts more easily. For example, if you build categories using Types as input, you could obtain a category **Fruit** with concepts such as **apple**, **pear**, **citrus fruits**, **orange** and so on. If you choose Type Patterns as input instead and select the pattern <Unknown> + <Positive>, for example, then you might get a category **fruit** + <Positive> with one or two kinds of fruit such as **fruit** + **tasty** and **apple** + **good**. This second result only shows 2 concept patterns because the other occurrences of fruit are not necessarily positively qualified. And while this might be good enough for your current text data, in longitudinal studies where you use different document sets, you may want to manually add in other descriptors such as **citrus fruit** + **positive** or use types. Using types alone as input will help you to find all possible fruit.

Techniques

Because every dataset is unique, the number of methods and the order in which you apply them may change over time. Since your text mining goals may be different from one set of data to the next, you may need to experiment with the different techniques to see which one produces the best results for the given text data.

You do not need to be an expert in these settings to use them. By default, the most common and average settings are already selected. Therefore, you can bypass the advanced setting dialogs and go straight to building your categories. Likewise, if you make changes here, you do not have to come back to the settings dialog each time since the latest settings are always retained.

Select either the linguistic or frequency techniques and click the Advanced Settings button to display the settings for the techniques selected. None of the automatic techniques will perfectly categorize your data; therefore we recommend finding and applying one or more automatic techniques that work well with your data. You cannot build using linguistic and frequency techniques simultaneously.

- Advanced linguistic techniques. See the topic [Advanced linguistic settings](#) for more information.
 - Advanced frequency techniques. See the topic [Advanced Frequency Settings](#) for more information.
 - [Advanced linguistic settings](#)
 - [About linguistic techniques](#)
 - [Advanced Frequency Settings](#)
-

Advanced linguistic settings

When you build categories, you can select from a number of advanced linguistic category building techniques such as *concept inclusion* and *semantic networks* (English text only). These techniques can be used individually or in combination with each other to create categories.

Keep in mind that because every dataset is unique, the number of methods and the order in which you apply them may change over time. Since your text mining goals may be different from one set of data to the next, you may need to experiment with the different techniques to see which one produces the best results for the given text data. None of the automatic techniques will perfectly categorize your data; therefore we recommend finding and applying one or more automatic techniques that work well with your data.

The following areas and fields are available within the Advanced Settings: Linguistics dialog box:

Input and Output

Category input Select from what the categories will be built:

- Unused extraction results. This option enables categories to be built from extraction results that are not used in any existing categories. This minimizes the tendency for records to match multiple categories and limits the number of categories produced.
- All extraction results. This option enables categories to be built using any of the extraction results. This is most useful when no or few categories already exist.

Category output Select the general structure for the categories that will be built:

- Hierarchical with subcategories. This option enables the creation of subcategories and sub-subcategories. You can set the depth of your categories by choosing the maximum number of levels (Maximum levels created field) that can be created. If you choose 3, categories

could contain subcategories and those subcategories could also have subcategories.

- Flat categories (single level only). This option enables only one level of categories to be built, meaning that no subcategories will be generated.

Grouping Techniques

Each of the techniques available is well suited to certain types of data and situations, but often it is helpful to combine techniques in the same analysis to capture the full range of documents or records. You may see a concept in multiple categories or find redundant categories.

Concept Inclusion. This technique builds categories by grouping multiterm concepts (compound words) based on whether they contain words that are subsets or supersets of a word in the other. For example, the concept **seat** would be grouped with **safety seat**, **seat belt**, and **seat belt buckle**. See the topic [Concept Inclusion](#) for more information.

Semantic Network. This technique begins by identifying the possible senses of each concept from its extensive index of word relationships and then creates categories by grouping related concepts. This technique is best when the concepts are known to the semantic network and are not too ambiguous. It is less helpful when text contains specialized terminology or jargon unknown to the network. In one example, the concept **granny smith apple** could be grouped with **gala apple** and **winesap apple** since they are siblings of the granny smith. In another example, the concept **animal** might be grouped with **cat** and **kangaroo** since they are hyponyms of **animal**. This technique is available for English text only in this release. See the topic [Semantic Networks](#) for more information.

Note: The Maximum search distance option is only available if you select Semantic Network.

Maximum search distance Select how far you want the techniques to search before producing categories. The lower the value, the fewer results you will get—however, these results will be less noisy and are more likely to be significantly linked or associated with each other. The higher the value, the more results you might get—however, these results may be less reliable or relevant. While this option is globally applied to all techniques, its effect is greatest on co-occurrences and semantic networks.

Prevent pairing of specific concepts. Select this checkbox to stop the process from grouping or pairing two concepts together in the output. To create or manage concept pairs, click [Manage Pairs...](#) See the topic [Managing Link Exception Pairs](#) for more information.

Generalize with wildcards where possible Select this option to allow the product to generate generic rules in categories using the asterisk wildcard. For example, instead of producing multiple descriptors such as **[apple tart + .]** and **[apple sauce + .]**, using wildcards might produce **[apple * + .]**. If you generalize with wildcards, you will often get exactly the same number of records or documents as you did before. However, this option has the advantage of reducing the number and simplifying category descriptors. Additionally, this option increases the ability to categorize more records or documents using these categories on new text data (for example, in longitudinal/wave studies).

Other Options for Building Categories

In addition to selecting the grouping techniques to apply, you can edit several other build options as follow:

Maximum number of top level categories created. Use this option to limit the number of categories that can be generated when you click the Build Categories button next. In some cases, you might get better results if you set this value high and then delete any of the uninteresting categories.

Minimum number of descriptors and/or subcategories per category. Use this option to define the minimum number of descriptors and subcategories a category must contain in order to be created. This option helps limit the creation of categories that do not capture a significant number of records or documents.

Allow descriptors to appear in more than one category When selected, this option allows descriptors to be used in more than one of the categories that will be built next. This option is generally selected since items commonly or "naturally" fall into two or more categories and allowing them to do so usually leads to higher quality categories. If you do not select this option, you reduce the overlap of records in multiple categories and depending on the type of data you have, this might be desirable. However, with most types of data, restricting descriptors to a single category usually results in a loss of quality or category coverage. For example, let's say you had the concept **car seat manufacturer**. With this option, this concept could appear in one category based on the text **car seat** and in another one based on **manufacturer**. But if this option is not selected, although you may still get both categories, the concept **car seat manufacturer** will only appear as a descriptor in the category it best matches based on several factors including the number of records in which **car seat** and **manufacturer** each occur.

Resolve duplicate category names by Select how to handle any new categories or subcategories whose names would be the same as existing categories. You can either merge the new ones (and their descriptors) with the existing categories with the same name. Alternatively, you can choose to skip the creation of any categories if a duplicate name is found in the existing categories.

- [Managing Link Exception Pairs](#)
-

Managing Link Exception Pairs

During category building, clustering, and concept mapping, the internal algorithms group words by known associations. To prevent two concepts from being paired, or linked together, you can turn on this feature in Build Categories Advanced Settings dialog, Build Clusters dialog, and Concept Map Index Settings dialog and click the Manage Pairs button.

In the resulting Manage Link Exceptions dialog, you can add, edit, or delete concept pairs. Enter one pair per line. Entering pairs here will prevent the pairing from occurring when building or extending categories, clustering, and concept mapping. Enter words exactly as you want them, for example the accented version of word is not equal to the unaccented version of the word.

For example, if you wanted to make sure that **hot dog** and **dog** are not grouped, you could add the pair as a separate line in the table.

Related information

- [Building categories](#)
 - [About linguistic techniques](#)
 - [Concept root derivation](#)
 - [Concept Inclusion](#)
 - [Semantic Networks](#)
 - [Co-occurrence Rules](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

About linguistic techniques

When you build or extend your categories, you can select from a number of advanced linguistic category building techniques including *concept inclusion* and *semantic networks* (English only). These techniques can be used individually or in combination with each other to create categories.

You do not need to be an expert in these settings to use them. By default, the most common and average settings are already selected. If you want, you can bypass this advanced setting dialog and go straight to building or extending your categories. Likewise, if you make changes here, you do not have to come back to the settings dialog each time since it will remember what you last used.

However, keep in mind that because every dataset is unique, the number of methods and the order in which you apply them may change over time. Since your text mining goals may be different from one set of data to the next, you may need to experiment with the different techniques to see which one produces the best results for the given text data. None of the automatic techniques will perfectly categorize your data; therefore we recommend finding and applying one or more automatic techniques that work well with your data.

The main automated linguistic techniques for category building are:

- Concept inclusion. This technique creates categories by taking a concept and finding other concepts that include it. See the topic [Concept Inclusion](#) for more information.
 - Semantic network. This technique begins by identifying the possible senses of each concept from its extensive index of word relationships and then creates categories by grouping related concepts. See the topic [Semantic Networks](#) for more information. This option is only available for English text.
 - [Concept root derivation](#)
 - [Concept Inclusion](#)
 - [Semantic Networks](#)
 - [Co-occurrence Rules](#)
-

Concept root derivation

The concept root derivation technique creates categories by taking a concept and finding other concepts that are related to it through analyzing whether any of the concept components are morphologically related. A component is a word. The technique attempts to group concepts by looking at the endings (suffixes) of each component in a concept and finding other concepts that could be derived from them. The idea is that when words are derived from each other, they are likely to share or be close in meaning. In order to identify the endings, internal language-specific rules are used. For example, the concept **opportunities to advance** would be grouped with the concepts **opportunity for advancement** and **advancement opportunity**.

You can use concept root derivation on any sort of text. By itself, it produces fairly few categories, and each category tends to contain few concepts. The concepts in each category are either synonyms or situationally related. You may find it helpful to use this algorithm even if you are building categories manually; the synonyms it finds may be synonyms of those concepts you are particularly interested in.

Note: You can prevent concepts from being grouped together by specifying them explicitly. See the topic [Managing Link Exception Pairs](#) for more information.

Term componentization and de-inflecting

When the concept root derivation or the concept inclusion techniques are applied, the terms are first broken down into components (words) and then the components are de-inflected. When a technique is applied, the concepts and their associated terms are loaded and split into

components based on separators, such as spaces, hyphens, and apostrophes. For example, the term **system administrator** is split into components such as **{administrator, system}**.

However, some parts of the original term may not be used and are referred to as stop words. In English, some of these ignorable components might include **a, and, as, by, for, from, in, of, on, or, the, to, and with**.

For example, the term **examination of the data** has the component set **{data, examination}**, and both **of** and **the** are considered ignorable. Additionally, component order is not in a component set. In this way, the following three terms could be equivalent: **cough relief for**

child, child relief from a cough, and relief of child

cough since they all have the same component set **{child, cough, relief}**. Each time a pair of terms are identified as being equivalent, the corresponding concepts are merged to form a new concept that references all of the terms.

Additionally, since the components of a term may be inflected, language-specific rules are applied internally to identify equivalent terms regardless of inflectional variation, such as plural forms. In this way, the terms **level of support** and **support levels** can be identified as equivalent since the de-inflected singular form would be **level**.

How concept root derivation works

After terms have been componentized and de-inflected (see previous section), the concept root derivation algorithm analyzes the component endings, or suffixes, to find the component root and then groups the concepts with other concepts that have the same or similar roots. The endings are identified using a set of linguistic derivation rules specific to the text language. For example, there is a derivation rule for English language text that states that a concept component ending with the suffix **ical** might be derived from a concept having the same root stem and ending with the suffix **ic**. Using this rule (and the de-inflection), the algorithm would be able to group the concepts **epidemiologic study** and **epidemiological studies**.

Since terms are already componentized and the ignorable components (for example, **in** and **of**) have been identified, the concept root derivation algorithm would also be able to group the concept **studies in epidemiology** with **epidemiological studies**.

The set of component derivation rules has been chosen so that most of the concepts grouped by this algorithm are synonyms: the concepts **epidemiologic studies, epidemiological studies, studies in epidemiology** are all equivalent terms. To increase completeness, there are some derivation rules that allow the algorithm to group concepts that are situationally related. For example, the algorithm can group concepts such as **empire builder** and **empire building**.

Concept Inclusion

The concept inclusion technique builds categories by taking a concept and, using lexical series algorithms, identifies concepts included in other concepts. The idea is that when words in a concept are a subset of another concept, it reflects an underlying semantic relationship. Inclusion is a powerful technique that can be used with any type of text.

This technique works well in combination with semantic networks but can be used separately. Concept inclusion may also give better results when the documents or records contain lots of domain-specific terminology or jargon. This is especially true if you have tuned the dictionaries beforehand so that the special terms are extracted and grouped appropriately (with synonyms).

How Concept Inclusion Works

Before the concept inclusion algorithm is applied, the terms are componentized and de-inflected. See the topic [Concept root derivation](#) for more information. Next, the concept inclusion algorithm analyzes the component sets. For each component set, the algorithm looks for another component set that is a subset of the first component set.

For example, if you have the concept **continental breakfast**, which has the component set **{breakfast, continental}**, and you have the concept **breakfast**, which has the component set **{breakfast}**, the algorithm would conclude that **continental breakfast** is a kind of **breakfast** and group these together.

In a larger example, if you have the concept **seat** in the Extraction Results pane and you apply this algorithm, then concepts such as **safety seat, leather seat, seat belt, seat belt buckle, infant seat carrier, and car seat laws** would also be grouped in that category.

Since terms are already componentized and the ignorable components (for example, **in** and **of**) have been identified, the concept inclusion algorithm would recognize that the concept **advanced spanish course** includes the concept **course in spanish**.

Note: You can prevent concepts from being grouped together by specifying them explicitly. See the topic [Managing Link Exception Pairs](#) for more information.

Related information

- [Building categories](#)
 - [Managing Link Exception Pairs](#)
 - [About linguistic techniques](#)
 - [Concept root derivation](#)
 - [Semantic Networks](#)
 - [Co-occurrence Rules](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Semantic Networks

In this release, the semantic networks technique is only available for English language text.

This technique builds categories using a built-in network of word relationships. For this reason, this technique can produce very good results when the terms are concrete and are not too ambiguous. However, you should not expect the technique to find many links between highly technical/specialized concepts. When dealing with such concepts, you may find the concept inclusion and concept root derivation techniques to be more useful.

How Semantic Network Works

The idea behind the semantic network technique is to leverage known word relationships to create categories of synonyms or hyponyms. A **hyponym** is when one concept is a sort of second concept such that there is a hierarchical relationship, also known as an ISA relationship. For example, if **animal** is a concept, then **cat** and **kangaroo** are hyponyms of **animal** since they are sorts of animals.

In addition to synonym and hyponym relationships, the semantic network technique also examines part and whole links between any concepts from the <Location> type. For example, the technique will group the concepts **normandy**, **provence**, and **france** into one category because Normandy and Provence are parts of France.

Semantic networks begin by identifying the possible senses of each concept in the semantic network. When concepts are identified as synonyms or hyponyms, they are grouped into a single category. For example, the technique would create a single category containing these three concepts: **eating apple**, **dessert apple**, and **granny smith** since the semantic network contains the information that: 1) **dessert apple** is a synonym of an **eating apple**, and 2) **granny smith** is a sort of **eating apple** (meaning it is a hyponym of **eating apple**).

Taken individually, many concepts, especially uniterms, are ambiguous. For example, the concept **buffet** can denote a sort of meal or a piece of furniture. If the set of concepts includes **meal**, **furniture** and **buffet**, then the algorithm is forced to choose between grouping **buffet** with **meal** or with **furniture**. Be aware that in some cases the choices made by the algorithm may not be appropriate in the context of a particular set of records or documents.

The semantic network technique can outperform concept inclusion with certain types of data. While both the semantic network and concept inclusion recognize that **apple pie** is a sort of **pie**, only the semantic network recognizes that **tart** is also a sort of **pie**.

Semantic networks will work in conjunction with the other techniques. For example, suppose that you have selected both the semantic network and inclusion techniques and that the semantic network has grouped the concept **teacher** with the concept **tutor** (because a tutor is a kind of teacher). The inclusion algorithm can group the concept **graduate tutor** with **tutor** and, as a result, the two algorithms collaborate to produce an output category containing all three concepts: **tutor**, **graduate tutor**, and **teacher**.

Options for Semantic Network

There are a number of additional settings that might be of interest with this technique.

- Change the Maximum search distance. Select how far you want the techniques to search before producing categories. The lower the value, the fewer results produced—however, these results will be less noisy and are more likely to be significantly linked or associated with each other. The higher the value, the more results you will get—however, these results may be less reliable or relevant.

For example, depending on the distance, the algorithm searches from **Danish pastry** up to **coffee roll** (its parent), then **bun** (grand parent) and on upwards to **bread**.

By reducing the search distance, this technique produces smaller categories that might be easier to work with if you feel that the categories being produced are too large or group too many things together.

Important! Additionally, we recommend that you do not apply the option Accommodate spelling errors for a minimum root character limit of (defined on the Expert tab of the node or in the Extract dialog box) for fuzzy grouping when using this technique since some false groupings can have a largely negative impact on the results.

Related information

- [Building categories](#)
 - [Managing Link Exception Pairs](#)
 - [About linguistic techniques](#)
 - [Concept root derivation](#)
 - [Concept Inclusion](#)
 - [Co-occurrence Rules](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Co-occurrence Rules

Co-occurrence rules enable you to discover and group concepts that are strongly related within the set of documents or records. The idea is that when concepts are often found together in documents and records, that co-occurrence reflects an underlying relationship that is probably of value in your category definitions. This technique creates co-occurrence rules that can be used to create a new category, extend a category, or as input to another category technique. Two concepts strongly co-occur if they frequently appear together in a set of records and rarely separately in any of the other records. This technique can produce good results with larger datasets with at least several hundred documents or records.

For example, if many records contain the words **price** and **availability**, these concepts could be grouped into a co-occurrence rule, (**price & available**). In another example, if the concepts **peanut butter**, **jelly**, **sandwich** appear more often together than apart, they would be grouped into a concept co-occurrence rule (**peanut butter & jelly & sandwich**).

Important! In earlier releases, co-occurrence and synonym rules were surrounded by square brackets. In this release, square brackets now indicate a text link analysis pattern result. Instead, co-occurrence and synonym rules will be encapsulated by parentheses such as (**speaker systems | speakers**).

How Co-occurrence Rules Works

This technique scans the documents or records looking for two or more concepts that tend to appear together. Two or more concepts strongly co-occur if they frequently appear together in a set of documents or records and if they seldom appear separately in any of the other documents or records.

When co-occurring concepts are found, a category rule is formed. These rules consist of two or more concepts connected using the & Boolean operator. These rules are logical statements that will automatically classify a document or record into a category if the set of concepts in the rule all co-occur in that document or record.

Options for Co-occurrence Rules

If you are using the co-occurrence rule technique, you can fine-tune several settings that influence the resulting rules:

- Change the Maximum search distance. Select how far you want the technique to search for co-occurrences. As you increase the search distance, the minimum similarity value required for each co-occurrence is lowered; as a result, many co-occurrence rules may be produced, but those which have a low similarity value will often be of little significance. As you reduce the search distance, the minimum required similarity value increases; as a result, fewer co-occurrence rules are produced, but they will tend to be more significant (stronger).
- Minimum number of documents. The minimum number of records or documents that must contain a given pair of concepts for it to be considered as a co-occurrence; the lower you set this option, the easier it is to find co-occurrences. Increasing the value results in fewer, but more significant, co-occurrences. As an example, suppose that the concepts "apple" and "pear" are found together in 2 records (and that neither of the two concepts occurs in any other records). With Minimum number of documents set to 2 (the default), the co-occurrence technique will create a category rule (apple and pear). If the value is raised to 3, the rule will no longer be created.

Note: With small datasets (< 1000 responses) you may not find any co-occurrences with the default settings. If so, try increasing the search distance value.

Note: You can prevent concepts from being grouped together by specifying them explicitly. See the topic [Managing Link Exception Pairs](#) for more information.

Related information

- [Building categories](#)
 - [Managing Link Exception Pairs](#)
 - [About linguistic techniques](#)
 - [Concept root derivation](#)
 - [Concept Inclusion](#)
 - [Semantic Networks](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Advanced Frequency Settings

You can build categories based on a straightforward and mechanical frequency technique. With this technique, you can build one category for each item (type, concept, or pattern) that was found above a given record or document count. Additionally, you can build a single category for all of the less frequently occurring items. By count, we refer to the number of records or documents containing the extracted concept (and any of its synonyms), type, or pattern in question as opposed to the total number of occurrences in the entire text.

Grouping frequently occurring items can yield interesting results, since it may indicate a common or significant response. The technique is very useful on the unused extraction results after other techniques have been applied. Another application is to run this technique immediately after extraction when no other categories exist, edit the results to delete uninteresting categories, and then extend those categories so that they match even more records or documents. See the topic [Extending categories](#) for more information.

Instead of using this technique, you could sort the concepts or concept patterns by descending number of records or documents in the Extraction Results pane and then drag and drop the top ones into the Categories pane to create the corresponding categories.

The following fields are available within the Advanced Settings: Frequencies dialog box:

Generate category descriptors at. Select the kind of input for descriptors. See the topic [Building categories](#) for more information.

- Concepts level. Selecting this option means that concepts or concept patterns frequencies will be used. Concepts will be used if types were selected as input for category building and concept patterns are used, if type patterns were selected. In general, applying this technique to the concept level will produce more specific results, since concepts and concept patterns represent a lower level of measurement.
- Types level. Selecting this option means that type or type patterns frequencies will be used. Types will be used if types were selected as input for category building and type patterns are used, if type patterns were selected. Applying this technique to the type level allows you to obtain a quick view regarding the kind of information present given.

Minimum doc. count for items to have their own category. This option allows you to build categories from frequently occurring items. This option restricts the output to only those categories containing a descriptor that occurred in at least X number of records or documents, where X is the value to enter for this option.

Group all remaining items into a category called. This option allows you to group all concepts or types occurring infrequently into a single 'catch-all' category with the name of your choice. By default, this category is named *Other*.

Category input. Select the group to which to apply the techniques:

- Unused extraction results. This option enables categories to be built from extraction results that are not used in any existing categories. This minimizes the tendency for records to match multiple categories and limits the number of categories produced.
- All extraction results. This option enables categories to be built using any of the extraction results. This is most useful when no or few categories already exist.

Resolve duplicate category names by. Select how to handle any new categories or subcategories whose names would be the same as existing categories. You can either merge the new ones (and their descriptors) with the existing categories with the same name. Alternatively, you can choose to skip the creation of any categories if a duplicate name is found in the existing categories.

Related information

- [Building categories](#)
 - [Advanced linguistic settings](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Extending categories

Extending is a process through which descriptors are added or enhanced automatically to 'grow' existing categories. The objective is to produce a better category that captures related records or documents that were not originally assigned to that category.

The automatic grouping techniques you select will attempt to identify concepts, TLA patterns, and category rules related to existing category descriptors. These new concepts, patterns, and category rules are then added as new descriptors or added to existing descriptors. The grouping techniques for extending include *concept root derivation*, *concept inclusion*, *semantic networks* (English only), and *co-occurrence rules*. The Extend empty categories with descriptors generated from the category name method generates descriptors using the words in the category names, therefore, the more descriptive the category names, the better the results.

Note: The frequency techniques are not available when extending categories.

Extending is a great way to interactively improve your categories. Here are some examples of when you might extend a category:

- After dragging/dropping concept patterns to create categories in the Categories pane
- After creating categories by hand and adding simple category rules and descriptors
- After importing a predefined category file in which the categories had very descriptive names
- After refining the categories that came from the TAP you chose

You can extend a category multiple times. For example, if you imported a predefined category file with very descriptive names, you could extend using the Extend empty categories with descriptors generated from the category name option to obtain a first set of descriptors, and then extend those categories again. However, in other cases, extending multiple times may result in too generic a category if the descriptors are extended wider and wider. Since the build and extend grouping techniques use similar underlying algorithms, extending directly after building categories is unlikely to produce more interesting results.

Tip:

- If you attempt to extend and do not want to use the results, you can always undo the operation (Edit > Undo) immediately after having extended.
- Extending can produce two or more category rules in a category that match exactly the same set of documents since rules are built independently during the process. If desired, you can review the categories and remove redundancies by manually editing the category description. See the topic [Editing Category Descriptors](#) for more information.

To extend categories

1. In the Categories pane, select the categories you want to extend.
 2. From the menus, choose Categories > Extend Categories. Unless you have chosen the option to never prompt, a message box appears.
 3. Choose whether you want to build now or edit the settings first.
- Click Extend Now to begin extending categories using the current settings. The process begins and a progress dialog appears.
 - Click Edit to review and modify the settings.

After attempting to extend, any categories for which new descriptors were found are flagged by the word Extended in the Categories pane so that you can quickly identify them. The Extended text remains until you either extend again, edit the category in another way, or clear these through the context menu.

Note: The maximum number of categories that can be displayed is 10,000. A warning is displayed if this number is reached or exceeded. If this happens you should change your Build or Extend Categories options to reduce the number of categories built.

Each of the techniques available when building or extending categories is well suited to certain types of data and situations, but often it is helpful to combine techniques in the same analysis to capture the full range of documents or records. In the interactive workbench, the concepts and types that were grouped into a category are still available the next time you build categories. This means that you may see a concept in multiple categories or find redundant categories.

The following areas and fields are available within the Extend Categories: Settings dialog box:

Extend with. Select what input will be used to extend the categories:

- Unused extraction results. This option enables categories to be built from extraction results that are not used in any existing categories. This minimizes the tendency for records to match multiple categories and limits the number of categories produced.
- All extraction results. This option enables categories to be built using any of the extraction results. This is most useful when no or few categories already exist.

Grouping techniques

For short descriptions of each of these techniques, see [Advanced linguistic settings](#). These techniques include:

- Concept root derivation
- Semantic network (English text only, and not used if the Generalize only option is selected.)
- Concept inclusion
- Co-occurrence and Minimum number of docs suboption.

A number of types are permanently excluded from the semantic networks technique since those types will not produce relevant results. They include <Positive>, <Negative>, <IP>, other non linguistic types, etc.

Maximum search distance Select how far you want the techniques to search before producing categories. The lower the value, the fewer results you will get—however, these results will be less noisy and are more likely to be significantly linked or associated with each other. The higher the value, the more results you might get—however, these results may be less reliable or relevant. While this option is globally applied to all techniques, its effect is greatest on co-occurrences and semantic networks.

Prevent pairing of specific concepts. Select this checkbox to stop the process from grouping or pairing two concepts together in the output. To create or manage concept pairs, click Manage Pairs... See the topic [Managing Link Exception Pairs](#) for more information.

Where possible: Choose whether to simply extend, generalize the descriptors using wildcards, or both.

- Extend and generalize. This option will extend the selected categories and then generalize the descriptors. When you choose to generalize, the product will create generic category rules in categories using the asterisk wildcard. For example, instead of producing multiple descriptors such as [apple tart + .] and [apple sauce + .], using wildcards might produce [apple * + .]. If you generalize with wildcards, you will often get exactly the same number of records or documents as you did before. However, this option has the advantage of reducing the number and simplifying category

descriptors. Additionally, this option increases the ability to categorize more records or documents using these categories on new text data (for example, in longitudinal/wave studies).

- Extend only. This option will extend your categories without generalizing. It can be helpful to first choose the Extend only option for manually-created categories and then extend the same categories again using the Extend and generalize option.
- Generalize only. This option will generalize the descriptors without extending your categories in any other way.

Note: Selecting this option disables the Semantic network option; this is because the Semantic network option is only available when a description is to be extended.

Other options for extending categories

In addition to selecting the techniques to apply, you can edit any of the following options:

Maximum number of items to extend a descriptor by. When extending a descriptor with items (concepts, types, and other expressions), define the maximum number of items that can be added to a single descriptor. If you set this limit to 10, then no more than 10 additional items can be added to an existing descriptor. If there are more than 10 items to be added, the techniques stop adding new items after the tenth is added.

Doing so can make a descriptor list shorter but doesn't guarantee that the most interesting items were used first. You may prefer to cut down the size of the extension without penalizing quality by using the Generalize with wildcards where possible option. This option only applies to descriptors that contain the Booleans & (AND) or ! (NOT).

Also extend subcategories. This option will also extend any subcategories below the selected categories.

Extend empty categories with descriptors generated from the category name. This method applies only to empty categories, which have 0 descriptors. If a category already contains descriptors, it will not be extended in this way. This option attempts to automatically create descriptors for each category based on the words that make up the name of the category. The category name is scanned to see if words in the name match any extracted concepts. If a concept is recognized, it is used to find matching concept patterns and these both are used to form descriptors for the category. This option produces the best results when the category names are both long and descriptive. This is a quick method for generating category descriptors, which in turn enable the category to capture records that contain those descriptors. This option is most useful when you import categories from somewhere else or when you create categories manually with long descriptive names.

Generate descriptors as. This option only applies if the preceding option is selected.

- Concepts. Choose this option to produce the resulting descriptors in the form of concepts, regardless of whether they have been extracted from the source text.
- Patterns. Choose this option to produce the resulting descriptors in the form of patterns, regardless of whether the resulting patterns or any patterns have been extracted.

Creating Categories Manually

In addition to creating categories using the automated category building techniques and the rule editor, you can also create categories manually. The following manual methods exist:

- Creating an empty category into which you will add elements one by one. See the topic [Creating New or Renaming Categories](#) for more information.
- Dragging terms, types, and patterns into the categories pane. See the topic [Creating Categories by Drag-and-Drop](#) for more information.
- [Creating New or Renaming Categories](#)
- [Creating Categories by Drag-and-Drop](#)

Related information

- [Categorizing text data](#)
- [The Categories Pane](#)
- [About Categories](#)
- [The Data Pane](#)
- [Building categories](#)
- [About linguistic techniques](#)
- [Extending categories](#)
- [Using Category Rules](#)
- [Using Text Analysis Packages](#)
- [Editing and Refining Categories](#)
- [Creating New or Renaming Categories](#)
- [Creating Categories by Drag-and-Drop](#)
- [Microsoft Internet Explorer settings for Help](#)

Creating New or Renaming Categories

You can create empty categories in order to add concepts and types into them. You can also rename your categories.

To Create a New Empty Category

1. Go to the Categories pane.
2. From the menus, choose Categories > Create Empty Category. The Category Properties dialog box opens.
3. Enter a name for this category in the Name field.
4. Click OK to accept the name and close the dialog box. The dialog box closes and a new category name appears in the pane.

You can now begin adding to this category. See the topic [Adding Descriptors to Categories](#) for more information.

To Rename a Category

1. Select a category and choose Categories > Rename Category. The Category Properties dialog box opens.
2. Enter a new name for this category in the Name field.
3. Click OK to accept the name and close the dialog box. The dialog box closes and a new category name appears in the pane.

Related information

- [Creating Categories Manually](#)
- [Creating Categories by Drag-and-Drop](#)
- [Microsoft Internet Explorer settings for Help](#)

Creating Categories by Drag-and-Drop

The drag-and-drop technique is manual and is not based on algorithms. You can create categories in the Categories pane by dragging:

- Extracted concepts, types, or patterns from the Extraction Results pane into the Categories pane.
- Extracted concepts from the Data pane into the Categories pane.
- Entire rows from the Data pane into the Categories pane. This will create a category made up of all of the extracted concepts and patterns contained in that row.

Note: The Extraction Results pane supports multiple selection to facilitate the dragging and dropping of multiple elements.

Important! You cannot drag and drop concepts from the Data pane that were not extracted from the text. If you want to force the extraction of a concept that you found in your data, you must add this concept to a type. Then run the extraction again. The new extraction results will contain the concept that you just added. You can then use it in your category. See the topic [Adding concepts to types](#) for more information.

To create categories using drag-and-drop:

1. From the Extraction Results pane or the Data pane, select one or more concepts, patterns, types, records, or partial records.
2. While holding the mouse button down, drag the element to an existing category or to the pane area to create a new category.
3. When you have reached the area where you would like to drop the element, release the mouse button. The element is added to the Categories pane. The categories that were modified appear with a special background color. This color is called the category feedback background. See the topic [Setting Options](#) for more information.

Note: The resulting category was automatically named. If you want to change a name, you can rename it. See the topic [Creating New or Renaming Categories](#) for more information.

If you want to see which records are assigned to a category, select that category in the Categories pane. The data pane is automatically refreshed and displays all of the records for that category.

Related information

- [Creating Categories Manually](#)
- [Creating New or Renaming Categories](#)
- [Microsoft Internet Explorer settings for Help](#)

Using Category Rules

You can create categories in many ways. One of these ways is to define category rules to express ideas. Category rules are statements that automatically classify documents or records into a category based on a logical expression using extracted concepts, types, and patterns as well as Boolean operators. For example, you could write an expression that means *include all records that contain the extracted concept embassy but not argentina in this category*.

While some category rules are produced automatically when building categories using grouping techniques such as *co-occurrence* and *concept root derivation* (Categories > Build Settings > Advanced Settings: Linguistics), you can also create category rules manually in the rule editor using your category understanding of the data and context. Each rule is attached to a single category so that each document or record matching the rule is then scored into that category.

Category rules help enhance the quality and productivity of your text mining results and further quantitative analysis by allowing you to categorize responses with greater specificity. Your experience and business knowledge might provide you with a specific understanding of your data and context. You can leverage this understanding to translate that knowledge into category rules to categorize your documents or records even more efficiently and accurately by combining extracted elements with Boolean logic.

The ability to create these rules enhances coding precision, efficiency, and productivity by allowing you to layer your business knowledge onto the product's extraction technology.

You can perform the following with rules:

- Create and test a rule. See the topic [Creating Category Rules](#) for more information.
- Edit or delete a rule. See the topic [Editing and Deleting Rules](#) for more information.

Note: For examples of how rules match text, see [Category Rule Examples](#)

- [Category Rule Syntax](#)
- [Using TLA Patterns in Category Rules](#)
- [Using Wildcards in Category Rules](#)
- [Category Rule Examples](#)
- [Creating Category Rules](#)
- [Editing and Deleting Rules](#)

Related information

- [Categorizing text data](#)
- [The Categories Pane](#)
- [About Categories](#)
- [The Data Pane](#)
- [Building categories](#)
- [About linguistic techniques](#)
- [Extending categories](#)
- [Creating Categories Manually](#)
- [Using Text Analysis Packages](#)
- [Editing and Refining Categories](#)
- [Category Rule Syntax](#)
- [Using TLA Patterns in Category Rules](#)
- [Using Wildcards in Category Rules](#)
- [Category Rule Examples](#)
- [Creating Category Rules](#)
- [Editing and Deleting Rules](#)
- [Microsoft Internet Explorer settings for Help](#)

Category Rule Syntax

While some category rules are produced automatically when building categories using grouping techniques such as *co-occurrence* and *concept root derivation* (Categories > Build Settings > Advanced Settings: Linguistics), you can also create category rules manually in the rule editor. Each rule is a descriptor of a single category; therefore, each document or record matching the rule is automatically scored into that category.

Note: For examples of how rules match text, see [Category Rule Examples](#)

When you are creating or editing a rule, you must have it open in the rule editor. You can add concepts, types, or patterns as well as use wildcards to extend the matches. When you use extracted concepts, types and patterns, you can benefit from finding all related concepts.

Important! To avoid common errors, we recommend dragging and dropping concepts directly from the Extraction Results pane, Text Link Analysis panes, or the Data pane into the rule editor or adding them via the context menus whenever possible.

When concepts, types, and patterns are recognized, an icon appears next to the text.

Table 1. Extraction icons

Icon	Description
	Extracted concept
	Extracted type
	Extracted pattern

Rule Syntax and Operators

The following table contains the characters with which you'll define your rule syntax. Use these characters along with the concepts, types, and patterns to create your rule.

Table 2. Supported syntax

Character	Description
&	The "and" boolean. For example, <code>a & b</code> contains both <code>a</code> and <code>b</code> such as: - <code>invasion & united states - 2016 & olympics - good & apple</code>
	The "or" boolean is inclusive, which means that if any or all of the elements are found, a match is made. For example, <code>a b</code> contains either <code>a</code> or <code>b</code> such as: - <code>attack france - condominium apartment</code>
! ()	The "not" boolean. For example, <code>!(a)</code> does not contain <code>a</code> . such as, <code>!(good & hotel)</code> , <code>assassination & !(austria)</code> , or <code>!(gold) & !(copper)</code>
*	A wildcard representing anything from a single character to a whole word depending how it is used. See the topic Using Wildcards in Category Rules for more information.
()	An expression delimiter. Any expression within the parenthesis is evaluated first.
+	The pattern connector used to form an order-specific pattern. When present, the square brackets must be used. See the topic Using TLA Patterns in Category Rules for more information.
[]	The pattern delimiter is required if you are looking to match based on an extracted TLA pattern inside of a category rule. The content within the brackets refers to TLA patterns and will never match concepts or types based on simple co-occurrence. If you did not extract this TLA pattern, then no match will be possible. See the topic Using TLA Patterns in Category Rules for more information. Do not use square brackets if you are looking to match concepts and types instead of patterns. Note: In older versions, co-occurrence and synonym rules generated by the category building techniques used to be surrounded by square brackets. In all new versions, square brackets now indicate the presence of a TLA pattern. Instead, rules produced by the co-occurrence technique and synonyms will be encapsulated in parentheses, such as <code>(speaker systems speakers)</code> .

The `&` and `|` operators are commutative such that `a & b = b & a` and `a | b = b | a`.

Escaping Characters with Backslash

If you have a concept that contains any character that is also a syntax character you must place a backslash in front of that character so that the rule is properly interpreted. The backslash (\) character is used to escape characters that otherwise have a special meaning. When you drag and drop into the editor, backslashing is done for you automatically.

The following rule syntax characters must be preceded by a backslash if you want it treated as it is rather than as rule syntax:

`\& \! \+ \< \> \(\) \[\] *`

For example, since the concept `r&d` contains the "and" operator (`&`), the backslash is required when it is typed into the rule editor, such as: `r\&d`.

Related information

- [Using Category Rules](#)
- [Using TLA Patterns in Category Rules](#)
- [Using Wildcards in Category Rules](#)
- [Category Rule Examples](#)
- [Creating Category Rules](#)
- [Editing and Deleting Rules](#)
- [Microsoft Internet Explorer settings for Help](#)

Using TLA Patterns in Category Rules

Text link analysis patterns can be explicitly defined in category rules to allow you to obtain even more specific and contextual results. When you define a pattern in a category rule, you are bypassing the more simple concept extraction results and only matching documents and records based on extracted text link analysis pattern results.

Important! In order to match documents using TLA patterns in your category rules, you must have run an extraction with text link analysis enabled. The category rule will look for the matches found during that process. If you did not choose to explore TLA results in the Model tab of

your Text Mining node, you can choose to enable TLA extraction in the extraction settings within the interactive session and then re-extract. See the topic [Extracting data](#) for more information.

Delimiting with square brackets. A TLA pattern must be surrounded by square brackets [] if you are using it inside of a category rule. The pattern delimiter is required if you are looking to match based on an extracted TLA pattern. Since category rules can contain, types, concepts, or patterns, the brackets clarify to the rule that the contents within the brackets refers to extracted TLA pattern. If you did not extract this TLA pattern, then no match will be possible. If you see a pattern without brackets such as `apple + good` in the Categories pane, this likely means that the pattern was added directly to the category outside of the category rule editor. For example, if you add a concept pattern directly to category from the text link analysis view, it will not appear with square brackets. However, when using a pattern within a category rule, you must encapsulate the pattern within the square brackets inside the category rule such as `[banana + ! (good)]`.

Using the + sign in patterns. In IBM® SPSS® Modeler Text Analytics, you can have up to a 6-part, or -slot, pattern. To indicate that the order is important, use the + sign to connect each element, such as `[company1 + acquired + company2]`. Here the order is important since it would change the meaning of which company was acquiring. Order is not determined by the sentence structure but rather by how the TLA pattern output is structured. For example, if you have the text "I love Paris" and you want to extract this idea, the TLA pattern is likely to be `[paris + like]` or `[<Location> + <Positive>]` rather than `[<Positive> + <Location>]` since the default opinion resources generally place opinions in the second position in 2 part patterns. So it can be helpful to use the pattern directly as a descriptor in your category to avoid issues. However, if you need to use a pattern as part of a more complex statement, pay particular attention to order of the elements within the patterns presented in the Text Link Analysis view since order plays a big role in whether a match can be found.

For example, let's say you had the two following sample texts the expression: "I like pineapple" and "I hate pineapple. However, I like strawberries". The expression `like & pineapple` would match both texts as it is a concept expression and not a text link rule (not enclosed in brackets). The expression `pineapple + like` matches only "I like pineapple" since in the second text, the word `like` is associated to `strawberries` instead.

Grouping with patterns. You can simplify your rules with your own patterns. Let's say you want to capture the following three expressions, `cayenne peppers + like`, `chili peppers + like`, and `peppers + like`. You can group them into a single category rule such as `[* peppers & like]`. If you had another expression `hot peppers + good`, you can group those four with a rule such as `[* peppers + <Positive>]`.

Order in patterns. In order to better organize output, the text link analysis rules supplied in the templates you installed with your product attempt to output basic patterns in the same order regardless of word order in the sentence. For example, if you had a record containing the text, "Good presentations." and another record containing "the presentations were good", both text are matched by the same rule and output in the same order as `presentation + good` in the concept pattern results rather than `presentation + good` and also `good + presentation`. And in two-slot pattern such as those in the example, the concepts assigned to types in the Opinions library will be presented last in the output by default such as `apple + bad`.

Table 1. Pattern syntax and boolean usage

Expression	Matches a document or record that
<code>[]</code>	Contains any TLA pattern. The pattern delimiter is required <i>in category rules</i> if you are looking to match based on an extracted TLA pattern. The content within the brackets refers to TLA patterns not simple concepts and types. If you did not extract this TLA pattern, then no match will be possible. If you wanted to create a rule that did not include any patterns, you could use <code>! ([])</code> .
<code>[a]</code>	Contains a pattern of which at least one element is <code>a</code> regardless of its position in the pattern. For example, <code>[deal]</code> can match <code>[deal + good]</code> or just <code>[deal + .]</code>
<code>[a + b]</code>	Contains a concept pattern. For example, <code>[deal + good]</code> . Note: If you only want to capture this pattern without adding any other elements, we recommend adding the pattern directly to your category rather than making a rule with it.
<code>[a + b + c]</code>	Contains a concept pattern. The + sign denotes that the order of the matching elements is important. For example, <code>[company1 + acquired + company2]</code> .
<code>[<A> +]</code>	Contains any pattern with type <code><A></code> in the first slot and type <code></code> in the second slot, and there are exactly two slots. The + sign denotes that the order of the matching elements is important. For example, <code>[<Budget> + <Negative>]</code> . Note: If you only want to capture this pattern without adding any other elements, we recommend adding the pattern directly to your category rather than making a rule with it.
<code>[<A> &]</code>	Contains any type pattern with type <code><A></code> and type <code></code> . For example, <code>[<Budget> & <Negative>]</code> . This TLA pattern will never be extracted; however, when written as such it is really equal to <code>[<Budget> + <Negative>] [<Negative> + <Budget>]</code> . The order of the matching elements is unimportant. Additionally, other elements might be in the pattern but it must have at least <code><Budget></code> and <code><Negative></code> .

Expression	Matches a document or record that
[a + .]	Contains a pattern where a is the only concept and there is nothing in any other slots for that pattern. For example, [deal + .] matches the concept pattern where the only output is the concept deal . If you added the concept deal as a category descriptor, you would get all records with deal as a concept including positive statements about a deal. However, using [deal + .] will match only those records pattern results representing deal and no other relationships or opinions and would not match deal + fantastic . Note: If you only want to capture this pattern without adding any other elements, we recommend adding the pattern directly to your category rather than making a rule with it.
[< A > + <>]	Contains a pattern where < A > is the only type. For example, [< Budget > + <>] matches the pattern where the only output is a concept of the type < Budget >. Note: You can use the <> to denote an empty type only when putting it after the pattern + symbol in type pattern such as [< Budget > + <>] but not [price + <>]. Note: If you only want to capture this pattern without adding any other elements, we recommend adding the pattern directly to your category rather than making a rule with it.
[a + !(b)]	Contains at least one pattern that includes the concept a but does not include the concept b . Must include at least one pattern. For example, [price + !(high)] or for types, [!(< Fruit > < Vegetable >) + < Positive >]
!([< A > & < B >])	Does not contain a specific pattern. For example, !([< Budget > & < Negative >]).

Note: For examples of how rules match text, see [Category Rule Examples](#)

Related information

- [Using Category Rules](#)
- [Category Rule Syntax](#)
- [Using Wildcards in Category Rules](#)
- [Category Rule Examples](#)
- [Creating Category Rules](#)
- [Editing and Deleting Rules](#)
- [Microsoft Internet Explorer settings for Help](#)

Using Wildcards in Category Rules

Wildcards can be added to concepts in rules in order to extend the matching capabilities. The asterisk * wildcard can be placed before and/or after a word to indicate how concepts can be matched. There are two types of wildcard uses:

- **Affix wildcards.** These wildcards immediately prefix or suffix without any space separating the string and the asterisk. For example, **operat*** could match *operat*, *operate*, *operates*, *operations*, *operational*, and so on.
- **Word wildcards.** These wildcards prefix or suffix a concept with a space between the concept and the asterisk. For example, * **operation** could match *operation*, *surgical operation*, *post operation*, and so on. Additionally, a word wildcard can be used along side an affix wildcard such as, * **operat*** *, which could match *operation*, *surgical operation*, *telephone operator*, *operatic aria*, and so on. As you can see in this last example, we recommend that wildcards be used with care so as not to cast the net too widely and capture unwanted matches.

Exceptions!

- A wildcard can never stand on its own. For example, (**apple** | *) would not be accepted.
- A wildcard can never be used to match type names. <**Negative***> will not match any type names at all.
- You cannot filter out certain types from being matched to concepts found through wildcards. The type to which the concept is assigned is used automatically.
- A wildcard can never be in the middle of a word sequence, whether it is end or beginning of a word (open* account) or a standalone component (open * account). You cannot use wildcards in type names either. For example, **word*** **word**, such as **apple*** **recipe**, will not match *applesauce recipe*, *apple pie*, *apple* and so on. In another example, **word *** **word**, such as **apple *** **toast**, will not match *apple cinnamon toast* or anything else at all since the asterisk appears between two other words. However, **apple *** would match *apple cinnamon toast*, *apple*, *apple pie* and so on.

Table 1. Wildcard usage

Expression	Matches a document or record that
* apple	Contains a concept that ends with letter written but may have any number of letters as a prefix. For example: * apple ends with the letters <i>apple</i> but can take a prefix such as: - apple - pineapple - crabapple

Expression	Matches a document or record that
apple*	Contains a concept that starts with letters written but may have any number of letters as a suffix. For example: apple* starts with the letters apple but can take a suffix or no suffix such as: - apple - applesauce - applejack For example, apple* & !(pear* quince) , which contains a concept that starts with the letters apple but not a concept starting with the letters pear or the concept quince , would NOT match: apple & quince but could match: - applesauce - apple & orange
product	Contains a concept that contains the letters written product , but may have any number of letters as either a prefix or suffix or both. For example: *product* could match: - product - byproduct - unproductive
* loan	Contains a concept that contains the word loan but may be a compound with another word placed before it. For example, * loan could match: - loan - car loan - home equity loan For example, [* delivery + <Negative>] contains a concept that ends in the word delivery in the first position and contains a type <Negative> in the second position could match the following concept patterns: - package delivery + slow overnight delivery + late
event *	Contains a concept that contains the word event but may be a compound followed by another word. For example, event * could match: - event - event location - event planning committee
* apple *	Contains a concept that might start with any word followed by the word apple possibly followed by another word. * means 0 or n, so it also matches apple . For example, * apple * could match: - gala applesauce - granny smith apple crumble - famous apple pie - apple For example, [* reservation * * + <Positive>], which contains a concept with the word reservation (regardless of where it is in the concept) in the first position and contains a type <Positive> in the second position could match the concept patterns: - reservation system + good - online reservation + good

Note: For examples of how rules match text, see [Category Rule Examples](#)

Related information

- [Using Category Rules](#)
- [Category Rule Syntax](#)
- [Using TLA Patterns in Category Rules](#)
- [Category Rule Examples](#)
- [Creating Category Rules](#)
- [Editing and Deleting Rules](#)
- [Microsoft Internet Explorer settings for Help](#)

Category Rule Examples

To help demonstrate how rules are matched to records differently based on the syntax used to express them, consider the following example.

Example Records

Imagine you had two records:

- **Record A:** “when I checked my wallet, I saw I was missing 5 dollars.”
- **Record B:** “\$5 was found at the picnic area, but the blanket was missing.”

The following two tables show what might be extracted for concepts and types as well as concept patterns and type patterns.

Concepts and Types Extracted From Example

Table 1. Example Extracted Concepts and Types

Extracted Concept	Concepts Typed As
wallet	<Unknown>
missing	<Negative>
USD5	<Currency>
blanket	<Unknown>
picnic area	<Unknown>

TLA Patterns Extracted From Example

Table 2. Example Extracted TLA Pattern Output

Extracted Concept Patterns	Extracted Type Patterns	From Record
picnic area + .	<Unknown> + <>	Record B
wallet + .	<Unknown> + <>	Record A
blanket + missing	<Unknown> + <Negative>	Record B
USD5 + .	<Currency> + <>	Record B
USD5 + missing	<Currency> + <Negative>	Record A

How Possible Category Rules Match

The following table contains some syntax that could be entered in the category rule editor. Not all rules here work and not all match the same records. See how the different syntax affects the records matched.

Table 3. Sample Rules

Rule Syntax	Result
<code>USD5 & missing</code>	Matches both records A and B since they both contain the extracted concept <code>missing</code> and the extracted concept <code>USD5</code> . This is equivalent to: <code>(USD5 & missing)</code>
<code>missing & USD5</code>	Matches both records A and B since they both contain the extracted concept <code>missing</code> and the extracted concept <code>USD5</code> . This is equivalent to: <code>(missing & USD5)</code>
<code>missing & <Currency></code>	Matches both records A and B since they both contain the extracted concept <code>missing</code> and a concept matching the type <code><Currency></code> . This is equivalent to: <code>(missing & <Currency>)</code>
<code><Currency> & missing</code>	Matches both records A and B since they both contain the extracted concept <code>missing</code> and a concept matching the type <code><Currency></code> . This is equivalent to: <code>(<Currency> & missing)</code>
<code>[USD5 + missing]</code>	Matches A but not B since record B did not produce any TLA pattern output containing <code>USD5 + missing</code> (see previous table). This is equivalent to the TLA pattern output: <code>USD5 + missing</code>
<code>[missing + USD5]</code>	Matches neither record A nor B since no extracted TLA pattern (see previous table) match the order expressed here with <code>missing</code> in the first position. This is equivalent to the TLA pattern output: <code>USD5 + missing</code>
<code>[missing & USD5]</code>	Matches A but not B since no such TLA pattern was extracted from record B. Using the character <code>&</code> indicates that order is unimportant when matching; therefore, this rule looks for a pattern match to either <code>[missing + USD5]</code> or <code>[USD5 + missing]</code> . Only <code>[USD5 + missing]</code> from record A has a match.
<code>[missing + <Currency>]</code>	Matches neither record A nor B since no extracted TLA pattern matched this order. This has no equivalent, since a TLA output is only based on terms <code>(USD5 + missing)</code> or on types <code>(<Currency> + <Negative>)</code> , but does not mix concepts and types.
<code>[<Currency> + <Negative>]</code>	Matches record A but not B since no TLA pattern was extracted from record B. This is equivalent to the TLA output: <code><Currency> + <Negative></code>
<code>[<Negative> + <Currency>]</code>	Matches neither record A nor B since no extracted TLA pattern matched this order. In the <code>Opinions</code> template, by default, when a <i>topic</i> is found with an <i>opinion</i> , the <i>topic</i> (<code><Currency></code>) occupies the first slot position and <i>opinion</i> (<code><Negative></code>) occupies the second slot position.

Related information

- [Using Category Rules](#)
- [Category Rule Syntax](#)
- [Using TLA Patterns in Category Rules](#)
- [Using Wildcards in Category Rules](#)
- [Creating Category Rules](#)
- [Editing and Deleting Rules](#)
- [Microsoft Internet Explorer settings for Help](#)

Creating Category Rules

When you are creating or editing a rule, you must have the rule open in the rule editor. You can add concepts, types, or patterns as well as use wildcards to extend the matches. When you use recognized concepts, types and patterns, you benefit since it will find all related concepts. For example, when you use a concept, all of its associated terms, plural forms, and synonyms are also matched to the rule. Likewise, when you use a type, all of its concepts are also captured by the rule.

You can open the rule editor by editing an existing rule or by right-clicking the category name and choosing Create Rule.

You can use context menus, drag-and-drop, or manually enter concepts, types, and patterns into the editor. Then combine these with Boolean operators (`&`, `!`, `()`, `|`) and brackets to form your rule expressions. To avoid common errors, we recommend dragging and dropping concepts directly from the Extraction Results pane or the Data pane into the rule editor. Pay close attention to the syntax of the rules to avoid errors. See the topic [Category Rule Syntax](#) for more information.

Note: For examples of how rules match text, see [Category Rule Examples](#).

To Create a Rule

1. If you have not yet extracted any data or your extraction is out of date, do so now. See the topic [Extracting data](#) for more information.
Note: If you filter an extraction in such a way that there are no longer any concepts visible, an error message is displayed when you attempt to create or edit a category rule. To prevent this, modify your extraction filter so that concepts are available.
2. In the Categories pane, select the category in which you want to add your rule.

3. From the menus choose Categories > Create Rule. The category rule editor pane opens in the window.
4. In the Rule Name field, enter a name for your rule. If you do not provide a name, the expression will be used as the name automatically. You can rename this rule later.
5. In the larger expression text field, you can:
 - Enter text directly in the field or drag-and-drop from another pane. Use only extracted concepts, types, and patterns. For example, if you enter the word **cats** but only the singular form, **cat**, appears in your Extraction Results pane, the editor will not be able to recognize **cats**. In this last case, the singular form might automatically include the plural, otherwise you could use a wildcard. See the topic [Category Rule Syntax](#) for more information.
 - Select the concepts, types, or patterns you want to add to rules and use the menus.
 - Add Boolean operators to link elements in your rule together. Use the toolbar buttons to add the "and" Boolean &, the "or" Boolean |, the "not" Boolean !(), parentheses (), and brackets for patterns [] to your rule.
6. Click the Test Rule button to verify that your rule is well-formed. See the topic [Category Rule Syntax](#) for more information. The number of documents or records found appears in parentheses next to the text Test result. To the right of this text, you can see the elements in your rule that were recognized or any error messages. If the graphic next to the type, pattern, or concept appears with a red question mark, this indicates that the element does not match any known extractions. If it does not match, then the rule will not find any records.
7. To test a part of your rule, select that part and click Test Selection.
8. Make any necessary changes and retest your rule if you found problems.
9. When finished, click Save & Close to save your rule again and close the editor. The new rule name appears in the category.

Related information

- [Using Category Rules](#)
- [Category Rule Syntax](#)
- [Using TLA Patterns in Category Rules](#)
- [Using Wildcards in Category Rules](#)
- [Category Rule Examples](#)
- [Editing and Deleting Rules](#)
- [Microsoft Internet Explorer settings for Help](#)

Editing and Deleting Rules

After you have created and saved a rule, you can edit that rule at any time. See the topic [Category Rule Syntax](#) for more information.

If you no longer want a rule, you can delete it.

To Edit Rules

1. In the Descriptors table in Category Definitions dialog box, select the rule.
2. From the menus choose Categories > Edit Rule or double-click the rule name. The editor opens with the selected rule.
3. Make any changes to the rule using extraction results and the toolbar buttons.
4. Retest your rule to make sure that it returns the expected results.
5. Click Save & Close to save your rule again and close the editor.

To Delete a Rule

1. In the Descriptors table in Category Definitions dialog box, select the rule.
2. From the menus, choose Edit > Delete. The rule is deleted from the category.

Related information

- [Using Category Rules](#)
- [Category Rule Syntax](#)
- [Using TLA Patterns in Category Rules](#)
- [Using Wildcards in Category Rules](#)
- [Category Rule Examples](#)
- [Creating Category Rules](#)
- [Microsoft Internet Explorer settings for Help](#)

Importing and Exporting Predefined Categories

If you have your own categories stored in an Microsoft Excel (*.xls, *.xlsx) file, you can import them into IBM® SPSS® Modeler Text Analytics . See the topic [Importing Predefined Categories](#) for more information.

You can also export the categories you have in an open interactive workbench session out to an Microsoft Excel (*.xls, *.xlsx) file. When you export your categories, you can choose to include or exclude some additional information such as descriptors and scores. See the topic [Exporting Categories](#) for more information.

If your predefined categories do not have codes or you want new codes, you can automatically generate a new set of codes for the set of categories in the categories pane by choosing Categories > Manage Categories > Autogenerate Codes from the menus. This will remove any existing codes and renumber them all automatically.

- [Importing Predefined Categories](#)
- [Exporting Categories](#)

Related information

- [Importing Predefined Categories](#)
- [Flat List Format](#)
- [Compact Format](#)
- [Indented Format](#)
- [Exporting Categories](#)
- [Microsoft Internet Explorer settings for Help](#)

Importing Predefined Categories

You can import your predefined categories into IBM® SPSS® Modeler Text Analytics . Before importing, make sure the predefined category file is in an Microsoft Excel (*.xls, *.xlsx) file and is structured in one of the supportive formats. You can also choose to have the product automatically detect the format for you. The following formats are supported:

- Flat list format: See the topic [Flat List Format](#) for more information.
- Compact format: See the topic [Compact Format](#) for more information.
- Indented format: See the topic [Indented Format](#) for more information.

To Import Predefined Categories

1. From the interactive workbench menus, choose Categories > Manage Categories > Import Predefined Categories. An Import Predefined Categories wizard is displayed.
2. From the Look In drop-down list, select the drive and folder in which the file is located.
3. Select the file from the list. The name of the file appears in the File Name text box.
4. Select the worksheet containing the predefined categories from the list. The worksheet name appears in the Worksheet field.
5. To begin choosing the data format, click Next.
6. Choose the format for your file or choose the option to allow the product to attempt to automatically detect the format. The autodetection works best on the most common formats.
 - Flat list format: See the topic [Flat List Format](#) for more information.
 - Compact format: See the topic [Compact Format](#) for more information.
 - Indented format: See the topic [Indented Format](#) for more information.
7. To define the additional import options, click Next. If you choose to have the format automatically detected, you are directed to the final step.
8. If one or more rows contain column headers or other extraneous information, select the row number from which you want to start importing in the Start import at row option. For example, if your category names begin on row 7, you must enter the number 7 for this option in order to import the file correctly.
9. If your file contains category codes, choose the option Contains category codes. Doing so helps the wizard properly recognize your data.
10. Review the color-coded cells and legend to make sure that the data has been correctly identified. Any errors detected in the file are shown in red and referenced below the format preview table. If the wrong format was selected, go back and choose another one. If you need to make corrections to your file, make those changes and restart the wizard by selecting the file again. You must correct all errors before you can finish the wizard.
11. To review the set of categories and subcategories that will be imported and to define how to create descriptors for these categories, click Next.
12. Review the set of categories that will be imported in the table. If you do not see the keywords you expected to see as descriptors, it may be that they were not recognized during the import. Make sure they are properly prefixed and appear in the correct cell.
13. Choose how you want to handle any pre-existing categories in your session.
 - Replace all existing categories. This option purges all existing categories and then the newly imported categories are used alone in their place.
 - Append to existing categories. This option will import the categories and merge any common categories with the existing categories. When adding to existing categories, you need to determine how you want any duplicates handled. One choice (option: Merge) is to merge any categories being imported with existing categories if they share a category name. Another choice (option: Exclude from import) is to prohibit the import of categories if one with the same name exists.
14. Import keywords as descriptors is an option to import the keywords identified in your data as descriptors for the associated category.

15. Extend categories by deriving descriptors is an option that will generate descriptors from the words that represent the name of the category, or subcategory, and/or the words that make up the annotation. If the words match extracted results, then those are added as descriptors to the category. This option produces the best results when the category names or annotations are both long and descriptive. This is a quick method for generating the category descriptors that enable the category to capture records that contain those descriptors.
 - From field allows you to select from what text the descriptors will be derived, the names or categories and subcategories, the words in the annotations, or both.
 - As field allows you to choose to create these descriptors in the form of concepts or TLA patterns. If TLA extraction has not taken place, the options of patterns are disabled in this wizard.
16. To import the predefined categories into the Categories pane, click Finish.

- [Flat List Format](#)
- [Compact Format](#)
- [Indented Format](#)

Related information

- [Importing and Exporting Predefined Categories](#)
 - [Flat List Format](#)
 - [Compact Format](#)
 - [Indented Format](#)
 - [Exporting Categories](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Flat List Format

In the flat list format, there is only one top level of categories without any hierarchy, meaning no subcategories or subnets. Category names are in a single column.

The following information can be contained in a file of this format:

- Optional **codes** column contains numerical values that uniquely identify each category. If you specify that the data file does contain codes (Contains category codes option in the Content Settings step), then a column containing unique codes for each category must exist in the cell directly to the left of category name. If your data does not contain codes, but you want to create some codes later, you can always generate codes later (Categories > Manage Categories > Autogenerate Codes).
- A **required category names** column contains all of the names of the categories. This column is required to import using this format.
- Optional **annotations** in the cell immediately to the right of the category name. This annotation consists of text that describes your categories/subcategories.
- Optional **keywords** can be imported as descriptors for categories. In order to be recognized, these keywords must exist in the cell directly below the associated category/subcategory name and the list of keywords must be prefixed by the underscore (_) character such as `_firearms, weapons / guns`. The keyword cell can contain one or more words used to describe each category. These words will be imported as descriptors or ignored depending on what you specify in the last step of the wizard. Later, descriptors are compared to the extracted results from the text. If a match is found, then that record or document is scored into the category containing this descriptor.

Table 1. Flat list format with codes, keywords, and annotations

Column A	Column B	Column C
Category code (<i>optional</i>)	Category name	Annotation
	<code>_Descriptor/keyword list (<i>optional</i>)</code>	

Related information

- [Importing and Exporting Predefined Categories](#)
 - [Importing Predefined Categories](#)
 - [Compact Format](#)
 - [Indented Format](#)
 - [Exporting Categories](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Compact Format

The compact format is structured similarly to the flat list format except that the compact format is used with hierarchical categories. Therefore, a code level column is required to define the hierarchical level of each category and subcategory.

The following information can be contained in a file of this format:

- A **required code level** column contains numbers that indicate the hierarchical position for the subsequent information in that row. For example, if values 1, 2 or 3 are specified and you have both categories and subcategories, then 1 is for categories, 2 is for subcategories, and 3 is for sub-subcategories. If you have only categories and subcategories, then 1 is for categories and 2 is for subcategories. And so on, until the desired category depth.
- Optional **codes** column contains values that uniquely identify each category. If you specify that the data file does contain codes (Contains category codes option in the Content Settings step), then a column containing unique codes for each category must exist in the cell directly to the left of category name. If your data does not contain codes, but you want to create some codes later, you can always generate codes later (Categories > Manage Categories > Autogenerate Codes).
- A **required category names** column contains all of the names of the categories and subcategories. This column is required to import using this format.
- Optional **annotations** in the cell immediately to the right of the category name. This annotation consists of text that describes your categories/subcategories.
- Optional **keywords** can be imported as descriptors for categories. In order to be recognized, these keywords must exist in the cell directly below the associated category/subcategory name and the list of keywords must be prefixed by the underscore (_) character such as `_firearms, weapons / guns`. The keyword cell can contain one or more words used to describe each category. These words will be imported as descriptors or ignored depending on what you specify in the last step of the wizard. Later, descriptors are compared to the extracted results from the text. If a match is found, then that record or document is scored into the category containing this descriptor.

Table 1. Compact format example with codes

Column A	Column B	Column C
Hierarchical code level	Category code (<i>optional</i>)	Category name
Hierarchical code level	Subcategory code (<i>optional</i>)	Subcategory name

Table 2. Compact format example without codes

Column A	Column B
Hierarchical code level	Category name
Hierarchical code level	Subcategory name

Related information

- [Importing and Exporting Predefined Categories](#)
- [Importing Predefined Categories](#)
- [Flat List Format](#)
- [Indented Format](#)
- [Exporting Categories](#)
- [Microsoft Internet Explorer settings for Help](#)

Indented Format

In the Indented file format, the content is hierarchical, which means it contains categories and one or more levels of subcategories. Furthermore, its structure is indented to denote this hierarchy. Each row in the file contains either a category or subcategory, but subcategories are indented from the categories and any sub-subcategories are indented from the subcategories, and so on. You can manually create this structure in Microsoft Excel or use one that was exported from another product and saved into an Microsoft Excel format.

- **Top level category codes and category names** occupy the columns A and B, respectively. Or, if no codes are present, then the category name is in column A.
- **Subcategory codes and subcategory names** occupy the columns B and C, respectively. Or, if no codes are present, then the subcategory name is in column B. The subcategory is a member of a category. You cannot have subcategories if you do not have top level categories.

Table 1. Indented structure with codes

Column A	Column B	Column C	Column D
Category code (<i>optional</i>)	Category name		
	Subcategory code (<i>optional</i>)	Subcategory name	
		Sub-subcategory code (<i>optional</i>)	Sub-subcategory name

Table 2. Indented structure without codes

Column A	Column B	Column C
Category name		
	Subcategory name	
		Sub-subcategory name

The following information can be contained in a file of this format:

- Optional **codes** must be values that uniquely identify each category or subcategory. If you specify that the data file does contain codes (Contains category codes option in the Content Settings step), then a unique code for each category or subcategory must exist in the cell directly to the left of category/subcategory name. If your data does not contain codes, but you want to create some codes later, you can always generate codes later (Categories > Manage Categories > Autogenerate Codes).
- A **required name** for each category and subcategory. Subcategories must be indented from categories by one cell to the right in a separate row.
- Optional **annotations** in the cell immediately to the right of the category name. This annotation consists of text that describes your categories/subcategories.
- Optional **keywords** can be imported as descriptors for categories. In order to be recognized, these keywords must exist in the cell directly below the associated category/subcategory name and the list of keywords must be prefixed by the underscore (_) character such as firearms, weapons / guns. The keyword cell can contain one or more words used to describe each category. These words will be imported as descriptors or ignored depending on what you specify in the last step of the wizard. Later, descriptors are compared to the extracted results from the text. If a match is found, then that record or document is scored into the category containing this descriptor.

Important! If you use a code at one level, you must include a code for each category and subcategory. Otherwise, the import process will fail.

Related information

- [Importing and Exporting Predefined Categories](#)
- [Importing Predefined Categories](#)
- [Flat List Format](#)
- [Compact Format](#)
- [Exporting Categories](#)
- [Microsoft Internet Explorer settings for Help](#)

Exporting Categories

You can also export the categories you have in an open interactive workbench session into an Microsoft Excel (*.xls, *.xlsx) file format. The data that will be exported comes largely from the current contents of the Categories pane or from the category properties. Therefore, we recommend that you score again if you plan to also export the Docs. score value.

Table 1. Category export options

Always gets exported...	Exported optionally...
<ul style="list-style-type: none"> • Category codes, if present • Category (and subcategory) names • Code levels, if present (<i>Flat/Compact</i> format) • Column headings (<i>Flat/Compact</i> format) 	<ul style="list-style-type: none"> • Docs. scores • Category annotations • Descriptor names • Descriptors counts

Important! When you export descriptors, they are converted to text strings and prefixed by an underscore. If you re-import into this product, the ability to distinguish between descriptors that are patterns, those that are category rules, and those that are plain concepts is lost. If you intend to reuse these categories in this product, we highly recommend making a text analysis package (TAP) file instead since the TAP format will preserve all descriptors as they are currently defined as well as all your categories, codes, and also the linguistic resources used. TAP files can be used in both IBM® SPSS® Modeler Text Analytics and IBM SPSS Text Analytics for Surveys . See the topic [Using Text Analysis Packages](#) for more information.

To Export Predefined Categories

1. From the interactive workbench menus, choose Categories > Manage Categories > Export Categories. An Export Categories wizard is displayed.
2. Choose the location and enter the name of the file that will be exported.
3. Enter a name for the output file in the File Name text box.
4. To choose the format into which you will export your category data, click Next.
5. Choose the format from the following:
 - Flat or Compact list format: See the topic [Flat List Format](#) for more information. Flat list contains no subcategories. See the topic [Compact Format](#) for more information. Compact list format contains hierarchical categories.
 - Indented format: See the topic [Indented Format](#) for more information.
6. To begin choosing the content to be exported and to review the proposed data, click Next.
7. Review the content for the exported file.
8. Select or unselect the additional content settings to be exported such as Annotations or Descriptor names.
9. To export the categories, click Finish.

Related information

- [Importing and Exporting Predefined Categories](#)
- [Importing Predefined Categories](#)

- [Flat List Format](#)
 - [Compact Format](#)
 - [Indented Format](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Using Text Analysis Packages

A text analysis package, also called a TAP, serves as a template for text response categorization. Using a TAP is an easy way for you to categorize your text data with minimal intervention since it contains the prebuilt category sets and the linguistic resources that are needed to code a vast number of records quickly and automatically. Using the linguistic resources, text data is analyzed and mined in order to extract key concepts. Based on key concepts and patterns that are found in the text, the records can be categorized into the category set you selected in the TAP. You can make your own TAP or update one.

A TAP is made up of the following elements:

- Category Set(s). A category set is essentially made up of predefined categories, category codes, descriptors for each category, and lastly, a name for the whole category set. Descriptors are linguistic elements (concepts, types, patterns, and rules) such as the term *cheap* or the pattern *good price*. Descriptors are used to define a category so that when the text matches any category descriptor, the document or record is put into the category.
- Linguistic Resources. Linguistic resources are a set of libraries and advanced resources that are tuned to extract key concepts and patterns. These extraction concepts and patterns, in turn, are used as the descriptors that enable records to be placed into a category in the category set.

The following tasks are possible with text analysis packages.

- Make text analysis packages. See [Making Text Analysis Packages](#) for more information.
- Load text analysis packages. Or you can load an SPSS® Text Analytics for Surveys project (.tas), which will be converted to a text analysis package. See [Loading Text Analysis Packages](#) for more information.
- Update text analysis packages. See [Updating Text Analysis Packages](#) for more information.

After you select the TAP and choose a category set, SPSS Modeler Text Analytics can extract and categorize your records.

Note: TAPs can be created and used interchangeably between SPSS Text Analytics for Surveys and SPSS Modeler Text Analytics . However, note that scoring on rules might be different in SPSS Modeler Text Analytics depending on whether you load a text analysis package (TAP) from SPSS Modeler Text Analytics directly, or whether you load a TAP from IBM® SPSS Text Analytics for Surveys . We recommend that you use TAPs that are made within SPSS Modeler Text Analytics ; this is because TAPs that are made in IBM SPSS Text Analytics for Surveys might be created by using a different version of the linguistic resources.

- [Making Text Analysis Packages](#)
 - [Loading Text Analysis Packages](#)
 - [Updating Text Analysis Packages](#)
-

Making Text Analysis Packages

Whenever you have a session with at least one category and some resources, you can make a text analysis package (TAP) from the contents of the open interactive workbench session. The set of categories and descriptors (concepts, types, rules or TLA pattern outputs) can be made into a TAP along with all of the linguistic resources open in the resource editor.

You can see the language for which the resources were created. The language is set in the Advanced Resources tab of the Template Editor or Resource Editor.

To Make a Text Analysis Package

1. From the menus, choose File > Text Analysis Packages > Make Package. The Make Package dialog appears.
2. Browse to the directory in which you will save the TAP. By default, TAPs are saved into the \TAP subdirectory of the product installation directory.
3. Enter a name for the TAP in the File Name field.
4. Enter a label in the Package Label field. When you enter a file name, this name automatically appears as the label but you can change this label.
5. To exclude a category set from the TAP, unselect the Include checkbox. Doing so will ensure that it is not added to your package. By default, one category set per question is included in the TAP. There must always be at least one category set in the TAP.
6. Rename any category sets. The New Category Set column contains generic names by default, which are generated by adding the `cat_` prefix to the text variable name. A single click in the cell makes the name editable. Enter or a click elsewhere applies the rename. If you rename a category set, the name changes in the TAP only and does not change the variable name in the open session.
7. Reorder the category sets if desired using the arrow keys to the right of the category set table.
8. Click Save to make the text analysis package. The dialog box closes.

Related information

- [Using Text Analysis Packages](#)
 - [Loading Text Analysis Packages](#)
 - [Updating Text Analysis Packages](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Loading Text Analysis Packages

When configuring a text mining modeling node, you must specify the resources that will be used during extraction. Instead of choosing a resource template, you can select a text analysis package (TAP) or an SPSS® Text Analytics for Surveys project (.tas) in order to copy not only its resources but also a category set into the node. If you select a .tas file, it will be converted to a TAP.

TAPs are most interesting when creating a category model interactively since you can use the category set as a starting point for categorization. When you run the stream, the interactive workbench session launches and this set of categories appears in the Categories pane. In this way, you score your documents and records immediately using these categories and then continue to refine, build, and extend these categories until they satisfy your needs. See [Methods and Strategies for Creating Categories](#) for more information.

Beginning in version 14, you can also see the language for which the resources in the TAP were defined when you click Load and choose the TAP.

To load a TAP or a TAS

1. Edit the Text Mining modeling node.
 2. On the Models tab, choose *Text analysis package* in the Copy Resources From section.
 3. Click Load. The Load Text Analysis Package dialog opens.
 4. Browse to the location of the TAP or the SPSS Text Analytics for Surveys project (.tas) containing the resources and category set you want to copy into the node. By default, they're saved to the \TAP subdirectory of your product installation directory.
 5. Enter a name for the TAP in the File Name field. The label is displayed automatically.
 6. Select the category set you want to use. This is the set of categories that will appear in the interactive workbench session. You can then tweak and improve these categories manually or using the Build or Extend categories options.
 7. Click Load to copy the contents of the text analysis package or SPSS Text Analytics for Surveys project into the node. The dialog box closes. When the contents are loaded, they're copied into the node; therefore, any changes you make to the external resources and categories will not be reflected unless you explicitly update it and reload it.
-

Updating Text Analysis Packages

If you make improvements to a category set, linguistic resources, or make a whole new category set, you can update a text analysis package (TAP) to make it easier to reuse these improvements later. To do so, you must be in the open session containing the information you want to put in the TAP. When you update, you can choose to append category sets, replace resources, change the package label, or rename/reorder category sets.

To update a text analysis package

1. From the menus, choose File > Text Analysis Packages > Update Package. The Update Package dialog appears.
2. Browse to the directory containing the text analysis package you want to update.
3. Enter a name for the TAP in the File Name field.
4. To replace the linguistic resources inside the TAP with those in the current session, select the Replace the resources in this package with those in the open session option. It generally makes sense to update the linguistic resources since they were used to extract the key concepts and patterns used to create the category definitions. Having the most recent linguistic resources ensures that you get the best results in categorizing your records. If you do not select this option, the linguistic resources that were already in the package are kept unchanged.
5. To update only the linguistic resources, make sure that you select the Replace the resources in this package with those in the open session option and select only the current category sets that were already in the TAP.
6. To include the new category set from the open session into the TAP, select the checkbox for each category set to be added. You can add one, multiple or none of the category sets.
7. To remove category sets from the TAP, deselect the corresponding Include checkbox. You might choose to remove a category set that was already in the TAP since you are adding an improved one. To do so, deselect the Include checkbox for the corresponding category set in the Current Category Set column. There must always be at least one category set in the TAP.
8. Rename category sets if needed. A single click in the cell makes the name editable. Enter or a click elsewhere applies the rename. If you rename a category set, the name changes in the TAP only and does not change the variable name in the open session. If two category sets have the same name, the names will appear in red until you correct the duplicate.

9. To create a new package with the session contents merged with the contents of the selected TAP, click Save As New. The Save As Text Analysis Package dialog appears. See following instructions.
10. Click Update to save the changes you made to the selected TAP.

To save a text analysis package

1. Browse to the directory in which you will save the TAP file. By default, TAP files are saved into the \TAP subdirectory of the installation directory.
 2. Enter a name for the TAP file in the File name field.
 3. Enter a label in the Package label field. When you enter a file name, this name is automatically used as the label. However, you can rename this label. You must have a label.
 4. Click Save to create the new package.
-

Editing and Refining Categories

Once you create some categories, you will invariably want to examine them and make some adjustments. In addition to refining the linguistic resources, you should review your categories by looking for ways to combine or clean up their definitions as well as checking some of the categorized documents or records. You can also review the documents or records in a category and make adjustments so that categories are defined in such a way that nuances and distinctions are captured.

You can use the built-in, automated, category-building techniques to create your categories; however, you are likely to want to perform a few tweaks to these categories. After using one or more technique, a number of new categories appear in the window. You can then review the data in a category and make adjustments until you are comfortable with your category definitions. See the topic [About Categories](#) for more information.

Here are some options for refining your categories:

- [Adding Descriptors to Categories](#)
- [Editing Category Descriptors](#)
- [Moving Categories](#)
- [Flattening Categories](#)
- [Merging or Combining Categories](#)
- [Forcing documents into categories](#)
- [Deleting Categories](#)

Related information

- [Adding Descriptors to Categories](#)
 - [Editing Category Descriptors](#)
 - [Moving Categories](#)
 - [Flattening Categories](#)
 - [Merging or Combining Categories](#)
 - [Deleting Categories](#)
 - [Categorizing text data](#)
 - [The Categories Pane](#)
 - [About Categories](#)
 - [The Data Pane](#)
 - [Building categories](#)
 - [About linguistic techniques](#)
 - [Extending categories](#)
 - [Creating Categories Manually](#)
 - [Using Category Rules](#)
 - [Using Text Analysis Packages](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Adding Descriptors to Categories

After using automated techniques, you will most likely still have extraction results that were not used in any of the category definitions. You should review this list in the Extraction Results pane. If you find elements that you would like to move into a category, you can add them to an existing or new category.

To Add a Concept or Type to a Category

1. From within the Extraction Results and Data panes, select the elements that you want to add to a new or existing category.
2. From the menus, choose Categories > Add to Category. The All Categories dialog box displays the set of categories. Select the category to which you want to add the selected elements. If you want to add the elements to a new category, select New Category. A new category appears in the Categories pane using the name of the first selected element.

Related information

- [Editing and Refining Categories](#)
- [Editing Category Descriptors](#)
- [Moving Categories](#)
- [Flattening Categories](#)
- [Merging or Combining Categories](#)
- [Deleting Categories](#)
- [Microsoft Internet Explorer settings for Help](#)

Editing Category Descriptors

Once you have created some categories, you can open each category to see all of the descriptors that make up its definition. Inside the Category Definitions dialog box, you can make a number of edits to your category descriptors. Also, if categories are shown in the category tree, you can also work with them there.

To Edit a Category

1. Select the category you want to edit in the Categories pane.
2. From the menus, choose View > Category Definitions. The Category Definitions dialog box opens.
3. Select the descriptor you want to edit and click the corresponding toolbar button.

The following table describes each toolbar button that you can use to edit your category definitions.

Table 1. Toolbar buttons and descriptions

Icons	Description
	Deletes the selected descriptors from the category.
	Moves the selected descriptors to a new or existing category.
	Moves the selected descriptors in the form of an & category rule to a category. See the topic Using Category Rules for more information.
	Moves each of the selected descriptors as its own new category
	Updates what is displayed in the Data pane and the Visualization pane according to the selected descriptors

Related information

- [Editing and Refining Categories](#)
- [Adding Descriptors to Categories](#)
- [Moving Categories](#)
- [Flattening Categories](#)
- [Merging or Combining Categories](#)
- [Deleting Categories](#)
- [Microsoft Internet Explorer settings for Help](#)

Moving Categories

If you want to place a category into another existing category or move descriptors into another category, you can move it.

To Move a Category

1. In the Categories pane, select the categories that you would like to move into another category.
2. From the menus, choose Categories > Move to Category. The menu presents a set of categories with the most recently created category at the top of the list. Select the name of the category to which you want to move the selected concepts.
 - If you see the name you are looking for, select it, and the selected elements are added to that category.
 - If you do not see it, select More to display the All Categories dialog box, and select the category from the list.

Related information

- [Editing and Refining Categories](#)
 - [Adding Descriptors to Categories](#)
 - [Editing Category Descriptors](#)
 - [Flattening Categories](#)
 - [Merging or Combining Categories](#)
 - [Deleting Categories](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Flattening Categories

When you have a hierarchical category structure with categories and subcategories, you can flatten your structure. When you flatten a category, all of the descriptors in the subcategories of that category are moved into the selected category and the now empty subcategories are deleted. In this way, all of the documents that used to match the subcategories are now categorized into the selected category.

To Flatten a Category

1. In the Categories pane, select a category (top level or subcategory) that you would like to flatten.
2. From the menus, choose Categories > Flatten Categories. The subcategories are removed and the descriptors are merged into the selected category.

Related information

- [Editing and Refining Categories](#)
 - [Adding Descriptors to Categories](#)
 - [Editing Category Descriptors](#)
 - [Moving Categories](#)
 - [Merging or Combining Categories](#)
 - [Deleting Categories](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Merging or Combining Categories

If you want to combine two or more existing categories into a new category, you can merge them. When you merge categories, a new category is created with a generic name. All of the concepts, types, and patterns used in the category descriptors are moved into this new category. You can later rename this category by editing the category properties.

To Merge a Category or Part of a Category

1. In the Categories pane, select the elements you would like to merge together.
2. From the menus, choose Categories > Merge Categories. The Category Properties dialog box is displayed in which you enter a name for the newly created category. The selected categories are merged into the new category as subcategories.

Related information

- [Editing and Refining Categories](#)
 - [Adding Descriptors to Categories](#)
 - [Editing Category Descriptors](#)
 - [Moving Categories](#)
 - [Flattening Categories](#)
 - [Deleting Categories](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Forcing documents into categories

Forcing documents into and out of categories enables you to override the category definitions created by the automatic category building techniques without changing the actual category definition. You may find that although the document contains terms that are used to define a

particular category, the document itself should not be in that category. In this case, you can force the document out of that category without having to remove the terms from the category definition.

Forcing is used in special cases where a document fits (or does not fit) a category, but for one reason or another (for example, it contains a particular term) is assigned (or not assigned) to that category. For example, this can occur when a respondent uses sarcasm in his or her response, such as "*The pizza was great. I am sure everyone loves burnt, cold pizza.*" Let's suppose that you had a category called **Pos: [<Food> + <Positive>]** to capture positive opinions regarding the food that a restaurant serves, and this response is assigned to that category. In this case, you might want to force this response out of the category.

To force in or out of categories

1. From within the Data pane, select the document that you want to force into or out of a particular category.
2. From the menus, choose **Categories > Force In** or **Categories > Force Out**. A submenu displays the list of categories from which you can select.
3. Select the category to which or from which you want to force this document. If you have created many categories, some may not be visible in the submenu.
 - In this case, choose **More** at the bottom of the submenu. The **All Categories** dialog box opens, in which you can select the category and click **OK** to apply the change.
 - If you want to force the document into a new category, select **Create Empty Category**. A new category appears in the category tree using a generic name.

Whenever a category contains one or more forced documents, a pseudo-category called either Force In or Force Out is displayed below the category name in the tree.

To clear a forced state

1. From within the Data pane, select the document that you no longer want to force into or out of a category.
2. From the menus, choose **Categories > Force In** to force in, or choose **Categories > Force Out** to force out. The categories in which the document is forced out of or into are preceded by a check mark.
3. Select the category in the submenu that is checked and for which you want to remove the force. The check mark is removed and the document is no longer forced.

To clear all forced states

1. From within the Data pane, select a record containing a Force In or Force Out.
2. From the menus, choose **Categories > Clear All > Force Ins** or **Categories > Clear All > Force Outs**. The forced state on the documents is cleared and they are no longer forced into or out of the categories.

Note: This feature is only available if your source text contains a unique ID. If the source text doesn't have a unique ID, you can add a Derive node between the source document and the Text Mining node. This feature only has an impact when running an interactive session. When you deploy the category model for non-interactive scoring, this piece of information is not preserved or used since it is based on a document ID.

Deleting Categories

If you no longer want to keep a category, you can delete it.

To Delete a Category

1. In the Categories pane, select the category or categories that you would like to delete.
2. From the menus, choose **Edit > Delete**.

Related information

- [Editing and Refining Categories](#)
- [Adding Descriptors to Categories](#)
- [Editing Category Descriptors](#)
- [Moving Categories](#)
- [Flattening Categories](#)
- [Merging or Combining Categories](#)
- [Microsoft Internet Explorer settings for Help](#)

Analyzing clusters

You can build and explore concept clusters in the Clusters view (View > Clusters). A *cluster* is a grouping of related concepts generated by clustering algorithms based on how often these concepts occur in the document/record set and how often they appear together in the same document, also known as *cooccurrence*. Each concept in a cluster cooccurs with at least one other concept in the cluster. The goal of clusters is to group concepts that co-occur together while the goal of categories is to group documents or records based on how the text they contain matches the descriptors (concepts, rules, patterns) for each category.

A good cluster is one with concepts that are strongly linked and cooccur frequently and with few links to concepts in other clusters. When working with larger datasets, this technique may result in significantly longer processing times.

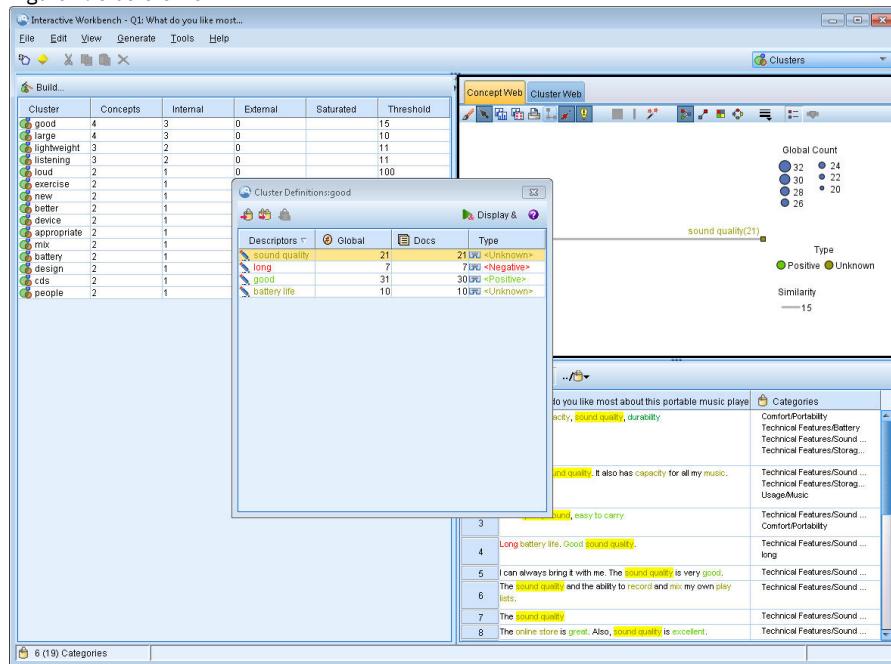
Clustering is a process that begins by analyzing a set of concepts and looking for concepts that cooccur often in documents. Two concepts that cooccur in a document are considered to be a concept pair. Next, the clustering process assesses the *similarity value* of each concept pair by comparing the number of documents in which the pair occur together to the number of documents in which each concept occurs. See the topic [Calculating Similarity Link Values](#) for more information.

Lastly, the clustering process groups similar concepts into clusters by aggregation and takes into account their link values and the settings defined in the Build Clusters dialog box. By aggregation, we mean that concepts are added or smaller clusters are merged into a larger cluster until the cluster is saturated. A cluster is *saturated* when additional merging of concepts or smaller clusters would cause the cluster to exceed the settings in the Build Clusters dialog box (number of concepts, internal links, or external links). A cluster takes the name of the concept within the cluster that has the highest overall number of links to other concepts within the cluster.

In the end, not all concept pairs end up together in the same cluster since there may be a stronger link in another cluster or saturation may prevent the merging of the clusters in which they occur. For this reason, there are both internal and external links.

- *Internal links* are links between concept pairs within a cluster. Not all concepts are linked to each other in a cluster. However, each concept is linked to at least one other concept inside the cluster.
- *External links* are links between concept pairs in separate clusters (a concept within one cluster and a concept outside in another cluster).

Figure 1. Clusters view



The Clusters view is organized into three panes, each of which can be hidden or shown by selecting its name from the View menu:

- Clusters pane You can build and manage your clusters in this pane. See the topic [Exploring Clusters](#) for more information.
- Visualization pane You can visually explore your clusters and how they interact in this pane. See the topic [Cluster Graphs](#) for more information.
- Data pane You can explore and review the text contained within documents and records that correspond to selections in the Cluster Definitions dialog box. See the topic [Cluster Definitions](#) for more information.
- [Building Clusters](#)
- [Exploring Clusters](#)

Building Clusters

When you first access the Clusters view, no clusters are visible. You can build the clusters through the menus (Tools > Build Clusters) or by clicking the Build... button on the toolbar. This action opens the Build Clusters dialog box in which you can define the settings and limits for

building your clusters.

Note: Whenever the extraction results no longer match the resources, this pane becomes yellow as does the Extraction Results pane. You can reextract to get the latest extraction results and the yellow coloring will disappear. However, each time an extraction is performed the Clusters pane is cleared, and you will have to rebuild your clusters. Likewise clusters are not saved from one session to another.

The following areas and fields are available within the Build Clusters dialog box:

Inputs

Inputs table Clusters are built from descriptors derived from certain types. In the table, you can select the types to include in the building process. The types that capture the most records or documents are preselected by default.

Concepts to cluster: Select the method of selecting the concepts you want to use for clustering. By reducing the number of concepts, you can speed up the clustering process. You can cluster using a number of top concepts, a percentage of top concepts, or using all the concepts:

- Number based on doc. count When you select Top number of concepts, enter the number of concepts to be considered for clustering. The concepts are chosen based on those that have the highest doc count value. Doc count is the number of documents or records in which the concept appears. The maximum value is 150,000.
- Percentage based on doc. count When you select Top percentage of concepts, enter the percentage of concepts to be considered for clustering. The concepts are chosen based on this percentage of concepts with the highest doc count value.

Output Limits

Maximum number of clusters to create This value is the maximum number of clusters to generate and display in the Clusters pane. During the clustering process, saturated clusters are presented before unsaturated ones, and therefore, many of the resulting clusters will be saturated. In order to see more unsaturated clusters, you can change this setting to a value greater than the number of saturated clusters.

Maximum concepts in a cluster This value is the maximum number of concepts a cluster can contain.

Minimum concepts in a cluster This value is the minimum number of concepts that must be linked in order to create a cluster.

Maximum number of internal links This value is the maximum number of internal links a cluster can contain. Internal links are links between concept pairs within a cluster.

Maximum number of external links This value is the maximum number of links to concepts outside of the cluster. External links are links between concept pairs in separate clusters.

Minimum link value This value is the smallest link value accepted for a concept pair to be considered for clustering. Link value is calculated using a similarity formula. See the topic [Calculating Similarity Link Values](#) for more information.

Prevent pairing of specific concepts. Select this checkbox to stop the process from grouping or pairing two concepts together in the output. To create or manage concept pairs, click Manage Pairs. See the topic [Managing Link Exception Pairs](#) for more information.

- [Calculating Similarity Link Values](#)

Related information

- [Calculating Similarity Link Values](#)
- [Analyzing clusters](#)
- [Exploring Clusters](#)
- [Cluster Definitions](#)
- [Microsoft Internet Explorer settings for Help](#)

Calculating Similarity Link Values

Knowing only the number of documents in which a concept pair cooccurs does not in itself tell you how similar the two concepts are. In these cases, the similarity value can be helpful. The similarity link value is measured using the cooccurrence document count compared to the individual document counts for each concept in the relationship. When calculating similarity, the unit of measurement is the number of documents (doc count) in which a concept or concept pair is found. A concept or concept pair is "found" in a document if it occurs *at least* once in the document. You can choose to have the line thickness in the Concept graph represent the similarity link value in the graphs.

The algorithm reveals those relationships that are strongest, meaning that the tendency for the concepts to appear together in the text data is much higher than their tendency to occur independently. Internally, the algorithm yields a similarity coefficient ranging from 0 to 1, where a value of 1 means that the two concepts always appear together and never separately. The similarity coefficient result is then multiplied by 100 and rounded to the nearest whole number. The similarity coefficient is calculated using the formula shown in the following figure.

Figure 1. Similarity coefficient formula

$$\text{similarity coefficient} = \frac{(C_{IJ})^2}{(C_I \times C_J)}$$

Where:

- C_I is the number of documents or records in which the concept I occurs.
- C_J is the number of documents or records in which the concept J occurs.
- C_{IJ} is the number of documents or records in which concept pair I and J cooccurs in the set of documents.

For example, suppose that you have 5,000 documents. Let I and J be extracted concepts and let IJ be a concept pair cooccurrence of I and J . The following table proposes two scenarios to demonstrate how the coefficient and link value are calculated.

Table 1. Concept frequencies example

Concept/Pair	Scenario A	Scenario B
Concept: I	Occurs in 20 docs	Occurs in 30 docs
Concept: J	Occurs in 20 docs	Occurs in 60 docs
Concept Pair: IJ	Cooccurs in 20 docs	Cooccurs in 20 docs
Similarity coefficient	1	0.22222
Similarity link value	100	22

In scenario A, the concepts I and J as well as the pair IJ occur in 20 documents, yielding a similarity coefficient of 1, meaning that the concepts always occur together. The similarity link value for this pair would be 100.

In scenario B, concept I occurs in 30 documents and concept J occurs in 60 documents, but the pair IJ occurs in only 20 documents. As a result, the similarity coefficient is 0.22222. The similarity link value for this pair would be rounded down to 22.

Related information

- [Building Clusters](#)
- [Microsoft Internet Explorer settings for Help](#)

Exploring Clusters

After you build clusters, you can see a set of results in the Clusters pane. For each cluster, the following information is available in the table:

- **Cluster.** This is the name of the cluster. Clusters are named after the concept with the highest number of internal links.
- **Concepts.** This is the number of concepts in the cluster. See the topic [Cluster Definitions](#) for more information.
- **Internal.** This is the number of internal links in the cluster. Internal links are links between concept pairs within a cluster.
- **External.** This is the number of external links in the cluster. External links are links between concept pairs when one concept is in one cluster and the other concept is in another cluster.
- **Sat.** If a symbol is present, this indicates that this cluster could have been larger but one or more limits would have been exceeded, and therefore, the clustering process ended for that cluster and is considered to be *saturated*. At the end of the clustering process, saturated clusters are presented before unsaturated ones and therefore, many of the resulting clusters will be saturated. In order to see more unsaturated clusters, you can change the Maximum number of clusters to create setting to a value greater than the number of saturated clusters or decrease the Minimum link value. See the topic [Building Clusters](#) for more information.
- **Threshold.** For all of the cooccurring concept pairs in the cluster, this is the lowest similarity link value of all in the cluster. See the topic [Calculating Similarity Link Values](#) for more information. A cluster with a high threshold value signifies that the concepts in that cluster have a higher overall similarity and are more closely related than those in a cluster whose threshold value is lower.

To learn more about a given cluster, you can select it and the visualization pane on the right will show two graphs to help you explore the cluster(s). See the topic [Cluster Graphs](#) for more information. You can also cut and paste the contents of the table into another application.

Whenever the extraction results no longer match the resources, this pane becomes yellow as does the Extraction Results pane. You can reextract to get the latest extraction results and the yellow coloring will disappear. However, each time an extraction is performed, the Clusters pane is cleared and you will have to rebuild your clusters. Likewise clusters are not saved from one session to another.

- [Cluster Definitions](#)

Related information

- [Analyzing clusters](#)
- [Building Clusters](#)
- [Cluster Definitions](#)
- [Microsoft Internet Explorer settings for Help](#)

Cluster Definitions

You can see all of the concepts inside a cluster by selecting it in the Clusters pane and opening the Cluster Definitions dialog box (View > Cluster Definitions).

All of the concepts in the selected cluster appear in the Cluster Definitions dialog box. If you select one or more concepts in the Cluster Definitions dialog box and click Display &, the Data pane will display all of the records or documents in which *all of the selected concepts appear together*. However, the Data pane does not display any text records or documents when you select a cluster in the Clusters pane. For general information on the Data pane, see [in](#).

Selecting concepts in this dialog box also changes the concept web graph. See the topic [Cluster Graphs](#) for more information. Similarly, when you select one or more concepts in the Cluster Definitions dialog box, the Visualization pane will show all of the external and internal links from those concepts.

Column Descriptions

Icons are shown so that you can easily identify each descriptor.

Table 1. Columns and Descriptor Icons

Columns	Description
Descriptors	The name of the concept.
 Global	Shows the number of times this descriptor appears in the entire dataset, also known as the global frequency.
 Docs	Shows the number of documents or records in which this descriptor appears, also known as the document frequency.
Type	Shows the type or types to which the descriptor belongs. If the descriptor is a category rule, no type name is shown in this column.

Toolbar Actions

From this dialog box, you can also select one or more concepts to use in a category. There are several ways to do this but it is most interesting to select concepts that cooccur in a cluster and add them as a category rule. See the topic [Co-occurrence Rules](#) for more information. You can use the toolbar buttons to add the concepts to categories.

Table 2. Toolbar buttons to add concepts to categories

Icons	Description
	Add the selected concepts to a new or existing category
	Add the selected concepts in the form of an & category rule to a new or existing category. See the topic Using Category Rules for more information.
	Add each of the selected concepts as its own new category
	Updates what is displayed in the Data pane and the Visualization pane according to the selected descriptors

Note: You can also add concepts to a type, as synonyms, or as exclude items using the context menus.

Related information

- [Analyzing clusters](#)
- [Building Clusters](#)
- [Exploring Clusters](#)
- [Microsoft Internet Explorer settings for Help](#)

Exploring Text Link Analysis

In the Text Link Analysis (TLA) view, you can explore text link analysis pattern results. Text link analysis is a pattern-matching technology that enables you to define pattern rules and compare these to actual extracted concepts and relationships found in your text.

For example, extracting ideas about an organization may not be interesting enough to you. Using TLA, you could also learn about the links between this organization and other organizations or the people within an organization. You can also use TLA to extract opinions on products or, for some languages, the relationships between genes.

Once you've extracted some TLA pattern results, you can review them in the Type and Concept Patterns panes of the Text Link Analysis view. See the topic [Type and Concept Patterns](#) for more information. You can further explore them in the Data or Visualization panes in this view. Possibly most importantly, you can add them to categories.

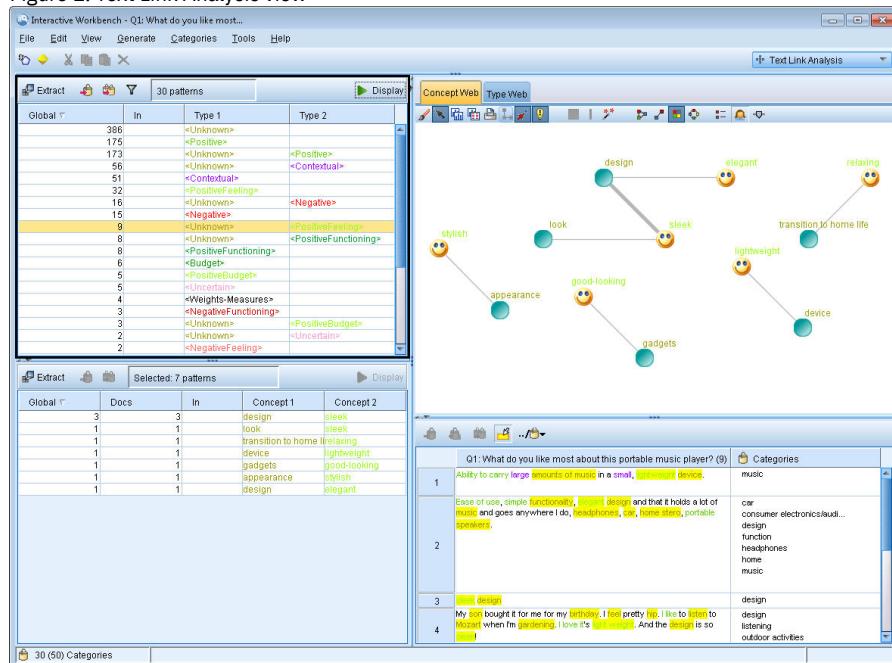
If you have not already chosen to do so, you can click Extract and choose Enable Text Link Analysis pattern extraction in the Extract Settings dialog box. See the topic [Extracting TLA Pattern Results](#) for more information.

There must be some TLA pattern rules defined in the resource template or libraries you are using in order to extract TLA pattern results. You can use the TLA patterns in certain resource templates shipped with IBM® SPSS® Modeler Text Analytics. The kind of relationships and patterns you can extract depend entirely on the TLA rules defined in your resources. You can define your own TLA rules. Patterns are made up of macros, word lists, and word gaps to form a Boolean query, or rule, that is compared to your input text. See the topic [About Text Link Rules](#) for more information.

Whenever a TLA pattern rule matches text, this text can be extracted as a pattern and restructured as output data. The results are then visible in the Text Link Analysis view panes. Each pane can be hidden or shown by selecting its name from the View menu:

- Type and Concept Patterns Panes. You can build and explore your patterns in these two panes. See the topic [Type and Concept Patterns](#) for more information.
- Visualization pane. You can visually explore how the concepts and types in your patterns interact in this pane. See the topic [Text Link Analysis Graphs](#) for more information.
- Data pane. You can explore and review text contained within documents and records that correspond to selections in another pane. See the topic [Data Pane](#) for more information.

Figure 1. Text Link Analysis view



- [Extracting TLA Pattern Results](#)
- [Type and Concept Patterns](#)
- [Filtering TLA Results](#)
- [Data Pane](#)
- [Type Reassignment Rules](#)

Extracting TLA Pattern Results

The extraction process results in a set of concepts and types, as well as Text Link Analysis (TLA) patterns, if enabled. If you extracted TLA patterns you can see those in the Text Link Analysis view. Whenever the extraction results are not in sync with the resources, the Patterns panes become yellow in color indicating that a reextraction would produce different results.

You have to choose to extract these patterns in the node setting or in the Extract dialog box using the option Enable Text Link Analysis pattern extraction. See the topic [Extracting data](#) for more information.

Note: There is a relationship between the size of your dataset and the time it takes to complete the extraction process. See the installation instructions for performance statistics and recommendations. You can always consider inserting a Sample node upstream or optimizing your machine's configuration.

To Extract Data

1. From the menus, choose Tools > Extract. Alternatively, click the Extract toolbar button.

2. Change any of the options you want to use. Keep in mind that the option Enable Text Link Analysis pattern extraction must be selected on this tab as well as having TLA rules in your template in order to extract TLA pattern results. See the topic [Extracting data](#) for more information.
3. Click Extract to begin the extraction process.

Once the extraction begins, the progress dialog box opens. If you want to abort the extraction, click Cancel. When the extraction is complete, the dialog box closes and the results appear in the pane. See the topic [Type and Concept Patterns](#) for more information.

Related information

- [Exploring Text Link Analysis](#)
 - [Type and Concept Patterns](#)
 - [Filtering TLA Results](#)
 - [Data Pane](#)
 - [Text Link Analysis Graphs](#)
 - [Concept Web Graph](#)
 - [Type Web Graph](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Type and Concept Patterns

Patterns are made up of two parts, a combination of concepts and types. Patterns are most useful when you are attempting to discover opinions about a particular subject or relationships between concepts. For example, extracting your competitor's product name may not be interesting enough to you. In this case, you can look at the extracted patterns to see if you can find examples where a document or record contains text expressing that the product is good, bad, or expensive.

Patterns can consist of up to six types or six concepts. For this reason, the rows in both patterns panes contain up to six slots, or positions. Each slot corresponds to an element's specific position in the TLA pattern rule as it is defined in the linguistic resources. In the interactive workbench, if a slot contains no values, it is not shown in the table. For example, if the longest pattern results contain no more than four slots, the last two are not shown. See the topic [About Text Link Rules](#) for more information.

When you extract pattern results, they are first grouped at the type level and then divided into concept patterns. For this reason, there are two different result panes: Type Patterns (upper left) and Concept Patterns (lower left). To see all concept patterns returned, select all of the type patterns. The bottom concept patterns pane will then display all concept patterns up to the maximum rank value (as defined in the Filter dialog box).

Type Patterns This pane presents pattern results consisting of one or more related types matching a TLA pattern rule. Type patterns are shown as <Organization> + <Location> + <Positive>, which might provide positive feedback about an organization in a specific location. The syntax is as follows:

```
<Type1> + <Type2> + <Type3> + <Type4> + <Type5> + <Type6>
```

Concept Patterns This pane presents the pattern results at the concept level for all of the type pattern(s) currently selected in the Type Patterns pane above it. Concept patterns follow a structure such as **hotel + paris + wonderful**. The syntax is as follows:

```
concept1 + concept2 + concept3 + concept4 + concept5 + concept6
```

When pattern results use less than the six maximum slots, only the necessary number of slots (or columns) are displayed. Any empty slots found between two filled slots are discarded such that the pattern <Type1>+<>+<Type2>+<>+<>+<> can be represented by <Type1>+<Type3>. For a concept pattern, this would be **concept1+.+concept2** (where . represents a null value).

Just as with the extraction results in the Categories and Concepts view, you can review the results here. If you see any refinements you would like to make to the types and concepts that make up these patterns, you make those in the Extraction Results pane in the Categories and Concepts view or directly in the Resource Editor and reextract your patterns. Whenever a concept, type, or pattern is used in a category definition as is or as part of a rule, a category or rule icon appears in the In column in the Pattern or Extraction Results table.

Note: If there are more results that can fit in the visible pane, you can use the controls at the bottom of the pane to move forwards and backwards through the results, or enter a page number to go to.

Related information

- [Exploring Text Link Analysis](#)
- [Extracting TLA Pattern Results](#)
- [Filtering TLA Results](#)
- [Data Pane](#)
- [Text Link Analysis Graphs](#)
- [Concept Web Graph](#)

- [Type Web Graph](#)
- [Microsoft Internet Explorer settings for Help](#)

Filtering TLA Results

When you are working with very large datasets, the extraction process could produce millions of results. For many users, this amount can make it more difficult to review the results effectively. You can, however, filter these results in order to zoom in on those that are most interesting. You can change the settings in the Filter dialog box to limit what patterns are shown. All of these settings are used together.

In the TLA view, the Filter dialog box contains the following areas and fields.

Filter by Frequency You can filter to display only those results with a certain global or document frequency value.

- Global frequency is the total number of times a pattern appears in the entire set of documents or records and is shown in the Global column.
- Document frequency is the total number of documents or records in which a pattern appears and is shown in the Docs column.

For example, if a pattern appeared 300 times in 500 records, we would say that this pattern has a global frequency of 300 and a document frequency of 500.

And by Match Text You can also filter to display only those results that match the rule you define here. Enter the set of characters to be matched in the Match text field, and select whether to look for this text within concept or type names by identifying the slot number or all of them. Then select the condition in which to apply the match (you do not need to use angled brackets to denote the beginning or end of a type name). Select either And or Or from the drop-down list so that the rule matches both statements or just one of them, and define the second text matching statement in the same manner as the first.

Table 1. Match text conditions

Condition	Description
Contains	Text is matched if the string occurs anywhere. (Default choice)
Starts with	Text is matched only if the concept or type starts with the specified text.
Ends with	Text is matched only if the concept or type ends with the specified text.
Exact Match	The entire string must match the concept or type name.

Results Displayed in Patterns Pane

Suppose you are using an English version of the software; here are some examples of how the results might be displayed on the Patterns pane toolbar based on the filters.

Figure 1. Filter results example 1



In this example, the toolbar shows that the number of patterns returned was limited because of the rank maximum specified in the filter. If a purple icon is present, this means that the maximum number of patterns was met. Hover over the icon for more information. See the preceding explanation of the And by Rank filter.

Figure 2. Filter results example 2



In this example, the toolbar shows results were limited using a match text filter (see magnifying glass icon). You can hover over the icon to see what the match text is.

To Filter the Results

1. From the menus, choose Tools > Filter. The Filter dialog box opens.
2. Select and refine the filters you want to use.
3. Click OK to apply the filters and see the new results.

Related information

- [Exploring Text Link Analysis](#)
- [Extracting TLA Pattern Results](#)
- [Type and Concept Patterns](#)
- [Data Pane](#)
- [Text Link Analysis Graphs](#)

- [Concept Web Graph](#)
 - [Type Web Graph](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Data Pane

As you extract and explore text link analysis patterns, you may want to review some of the data you are working with. For example, you may want to see the actual records in which a group of patterns were discovered. You can review records or documents in the Data pane, which is located in the lower right. If not visible by default, choose **View > Panes > Data** from the menus.

The Data pane presents one row per document or record corresponding to a selection in the view, up to a certain display limit. By default, the number of documents or records shown in the Data pane is limited in order to make it faster for you to see your data. However, you can adjust this in the Options dialog box. See [Options: Session Tab](#) for more information.

Note: If there are more results that can fit in the visible pane, you can use the controls at the bottom of the pane to move forwards and backwards through the results, or enter a page number to go to.

Displaying and refreshing the Data pane

The Data pane does not refresh its display automatically, because with larger datasets automatic data refreshing could take some time to complete. Therefore, whenever you select type or concept patterns in this view, you can click **Display** to refresh the contents of the Data pane.

Text Documents or Records

If your text data is in the form of records and the text is relatively short in length, the text field in the Data pane displays the text data in its entirety. However, when working with records and larger datasets, the text field column shows a short piece of the text and opens a Text Preview pane to the right to display more or all of the text of the record you have selected in the table. If your text data is in the form of individual documents, the Data pane shows the document's filename. When you select a document, the Text Preview pane opens with the selected document's text.

Colors and Highlighting

Whenever you display the data, concepts and descriptors found in those documents or records are highlighted in color to help you easily identify them in the text. The color coding corresponds to the types to which the concepts belong. You can also hover your mouse over color-coded items to display the concept under which it was extracted and the type to which it was assigned. Any text that was not extracted appears in black. Typically, these unextracted words are often connectors (*and* or *with*), pronouns (*me* or *they*), and verbs (*is*, *have*, or *take*).

Data Pane Columns

While the text field column is always visible, you can also display other columns. To display other columns, choose **View > Data Pane** from the menus, and then select the column that you want to display in the Data pane. The following columns may be available for display:

- "Text field name" (#)/Documents. Adds a column for the text data from which concepts and type were extracted. If your data is in documents, the column is called Documents and only the document filename or full path is visible. To see the text for those documents you must look in the Text Preview pane. The number of rows in the Data pane is shown in parentheses after this column name. There may be times when not all documents or records are shown due to a limit in the Options dialog used to increase the speed of loading. If the maximum is reached, the number will be followed by - Max. See [Options: Session Tab](#) for more information.
- Categories. Lists each of the categories to which a record belongs. Whenever this column is shown, refreshing the Data pane may take a bit longer so as to show the most up-to-date information.
- Force In. Lists the categories into which you have forced a document. Documents can be forced into the category through the **Edit > Force In** menu selection. See [Forcing documents into categories](#) for more information.
- Force Out. Lists the categories from which you have removed a document. Documents can be forced out of a category through the **Edit > Force Out** menu selection. For example, this might be used when a respondent's sarcasm causes a response to be miscategorized. See [Forcing documents into categories](#) for more information.
- Category Counts. Lists the number of the categories to which a record belongs.
- Relevance Ranks. Provides a rank for each record in a single category. This rank shows how well the record fits into the category compared to the other records in that category. Select a category in the Categories pane (upper left pane) to see the rank. See [Category Relevance](#) for more information.
- Response Flags. Adds a column that shows any flags you may be using. Click inside this column to change the type of flag that you assign to documents. You might flag documents with a "complete" flag or an "important" flag, or remove flags. This is useful for reviewing the completeness of a category model. See [Flagging responses](#) for more information.
- [Flagging responses](#)

Flagging responses

To help you monitor your progress, you can mark documents using flags in the Data pane. This feature is only available if the source document contains a unique ID. If the source document doesn't contain a unique ID, you can add a Derive node between the source document and the Text Mining node.

There are many reasons why you might want to mark a document, including:

- To mark off documents that you have manually reviewed so that you know where to pick up later
- To mark off a document that you are unsure about how to handle

Once you mark a document with a flag, you can continue to work with the documents. They are purely for your own record-keeping. You can choose from the following flags:

Table 1. Flag descriptions

Flag	Description
🏁	Complete flag to denote documents that you deem finished.
🚩	Important flag to denote documents that you deem important.

To mark a document with a flag:

1. From within the Data pane, right-click the document that you want to mark.
2. From the context menu, choose View > Data Pane > Response Flags and then select the type of flag that you want to use (Important Flag or Complete Flag). The selected flag is assigned. If the Flag column in the Data pane is not visible, it appears.

To clear flags:

1. From within the Data pane, right-click the documents for which you want to remove a flag.
2. From the context menu, choose Mark Responses With > Clear Flags. The selected flags are removed.

Type Reassignment Rules

Type Reassignment Rules (TRRs) aim to transform a sequence of types, macros, and/or tokens into a new concept with a specific type. Specifically, they're used in Opinions templates to catch opinions with a change in polarity. For example, in the phrase "**not that bad**," the word "**bad**" is a *negative* opinion. But in this context, the real meaning is "**not bad**" – which is a *positive*.

Up until version 18.2, this change in polarity was managed by specific Text Link Analysis (TLA) rules:

Figure 1. TLA rules

Element	Quantity	Example Token
{ mAdvNeg mSupportNeg mSupportNegPart mMisNeg }	Exactly 1	it's just not
{ mSupport mAdverb mToo }	0 or 1	?
mEmpty	Between 0 and 5	?
{ Neg Contextual }	Exactly 1	bad

Concept 1	Type 1	Concept 2	Type 2	Concept 3	Type 3	Concept 4	Type 4
good (4)	Positive						

Because of the different opinion types (**Positive**, **PositiveAttitude**, **PositiveBudget**, **PositiveCompetence**, **PositiveFeeling**, **PositiveFunctioning**, **PositiveRecommendation**, **Negative**, **NegativeAttitude**, **NegativeBudget**, **NegativeCompetence**, **NegativeFeeling**, **NegativeFunctioning**, **NegativeRecommendation**, and **Contextual**), it involved writing specific TLA rules:

- For each type. For example:

```
"not + xxx + <NegativeBudget>" => "<PositiveBudget>"
```

or

```
"not + xxx + <PositiveAttitude>" => "<NegativeAttitude>"
```

- For many syntactic contexts. For example:

```
* topic + negation + opinion ("hotel wasn't good")
* negation + opinion + topic ("it was not a good hotel")
* negation + opinion ("not very good")
* topic + opinion + negation + opinion ("hotel was well-located but not that good")
```

```
* 2 topics + negation + opinion ("room and swimming pool weren't always clean")
* ...
```

Beginning with version 18.2, the new approach is to "catch" such sequences (any negation + any empty word + a specific opinion), select the words to appear in the new concept (a standardized negation – for instance, "not" – and the opinion), and define a type for this new concept (aka, "pseudo-term"). This new concept can then be used in TLA rules.

As a consequence, the following rule will match any sequence containing a topic followed by an opinion, whether the opinion is a term (**comfortable**) or a psuedo-term (**not economical**), regardless of specific opinion sub-type (Attitude, Budget, etc).

```
#@# Bed was extremely comfortable
[pattern(190)]
name=topic + opinion_190
value=$mTopic {$_mEmpty|$_mToo} {0,3} {$_mOpinionPos|$_mOpinionNeg|$_Contextual}
output(1)=$1$t#1\$_3$t#3
```

Along with changing polarity for opinions, you can also use TRRs to help fine-tune your dictionary. For example, let's suppose you have a type called **Anatomy** with body parts such as **heart**, **chest**, **breast**, and **adrenal gland** – and another type called **MedicalProcedures** with procedures such as **biopsy**, **needle**

biopsy, **MRI**, and **CT scan**. It would be nearly impossible to list all the medical procedures correctly associated with an organ. So you could create two TRRs to identify potential medical procedures as seen in the following figures. Then once extraction is performed, you can add a filter on the type **PotentialMedicalProcedures**, review the candidate terms, and then add them to the **MedicalProcedures** type.

Figure 2. TRR for anatomy + medical procedures

Element	Quantity	Example Token
1 \$g Anatomy	Exactly 1	
2 \$g MedicalProcedures	Exactly 1	

Concept	Type
\$g PotentialMedicalProcedures	

Figure 3. TRR for medical procedures + anatomy

Element	Quantity	Example Token
1 \$g MedicalProcedures	Exactly 1	scan
2 \$g Anatomy	Exactly 1	of

Concept	Type
scan (4) (1)	\$g MedicalProcedures

Syntax

```
#@# not that expensive
[typeReassignmentRule]
name=TRR_"not" NegativeBudget
value=$mAllNeg {$_mAdverb|$_mBe|$_mHave|$_mSupport|$_mDet|that|more|$_mQuant} {0,3} $NegativeBudget
output=not $3\$_PositiveBudget
```

- "name" must be unique (**TRR_"not"**). It can't be used in a macro or in a TLA rule. Only the type defined in the output can be used.
 - "value" is a sequence of elements to match. The elements can be types (**\$NegativeBudget**), macros (**\$mAllNeg**) or tokens (**more**). Some elements can be required, optional, or have a specific quantity.
 - "output" is a SINGLE pair of concept+type (**not \$3\\$_PositiveBudget**). Note that in the output you can use an available type (one that is defined in the template) or you can create a new type.
- The output type can also reference a matched element (for example, **#2**). This feature is especially useful when there is no change in type between the value and the output. For example:

```
#@# could not have been any more pleased
[typeReassignmentRule]
name=TRR_"couldn't be more" opinion
value=$mNotNeg {$_mOpinionPos|$_mOpinionNeg|$_Contextual}
output=$2\$_#2
```

As in TLA rules, a more specific TRR must be defined before a more generic one. To ensure all TRRs are defined in the correct order, you can use the Get Tokens feature to test each TRR in sequence. If a TRR doesn't match, but matches another definition, you can move it down (or up).

Special cases

In some cases, it is necessary to still have access to the individual elements of the sequence and not to a TRR. This typically concerns coordination over negation. In the phrase "not that fashionable or eyecatching," the coordination "or" doesn't allow for discovering that, in this context, "eyecatching" actually means "not eyecatching."

In this case, we recommend using a specific rule such as:

```
#@# not that fashionable or eyecatching
[pattern(263)]
name="not" + 2 Positive_263
value=($mAdvNeg|$mSupportNeg|$mMiscNeg) @{0,1} $PositiveFeeling or $PositiveFeeling
output(1)=not $3\tNegativeFeeling
output(2)=not $5\tNegativeFeeling
```

Although the first part of the rule `(($mAdvNeg | $mSupportNeg | $mMiscNeg) @{0,1} $PositiveFeeling or $PositiveFeeling)` may match a TRR, the TLA rule will have priority.

If you write a more generic rule such as the following example, the same restrictions that existed in version 18.1.1 and prior will still apply. The new concept created (pseudo-concept) may have an incorrect type (`<Negative>` instead of `<NegativeFeeling>`), and you may end up with a TLA concept with two different types. A workaround is to create the corresponding term (not `xxx`) with the correct type.

```
#@# not that fashionable or eyecatching
[pattern(263)]
name="not" + 2 Positive_263
value=($mAdvNeg|$mSupportNeg|$mMiscNeg) @{0,1} $mPos or $mPos
output(1)=not $3\tNegative
output(2)=not $5\tNegative
```

Advantages

- The main advantage of using TRRs is to have less TLA rules.
- A less obvious advantage is that TRRs ensure that a pseudo-term will mostly result in the correct type (but keep in mind the restriction mentioned previously). In the past, some "`not + positiveXXX`" were typed as `Negative` instead of `NegativeXXX`, because some specific TLA rules were missing.
- If a user adds a specific opinion type (for example, `NegativeNoise`), there is no need to replicate specific TLA rules to invert polarity. The user just needs to create the relevant TRR.

Visualizing Graphs

The Categories and Concepts view, Clusters view, and Text Link Analysis view all have a visualization pane in the upper right corner of the window. You can use this pane to visually explore your data. The following graphs and charts are available.

- **Categories and Concepts view.** This view has three graphs and charts: *Category Bar*, *Category Web*, and *Category Web Table*. In this view, the graphs are only updated when you click Display. See the topic [Category Graphs and Charts](#) for more information.
- **Clusters view.** This view has two web graphs: *Concept Web Graph* and *Cluster Web Graph*. See the topic [Cluster Graphs](#) for more information.
- **Text Link Analysis view.** This view has two web graphs: *Concept Web Graph* and *Type Web Graph*. See the topic [Text Link Analysis Graphs](#) for more information.
- [Category Graphs and Charts](#)
- [Cluster Graphs](#)
- [Text Link Analysis Graphs](#)
- [Using Graph Toolbars and Palettes](#)

Related information

- [Category Graphs and Charts](#)
- [Cluster Graphs](#)
- [Text Link Analysis Graphs](#)
- [Using Graph Toolbars and Palettes](#)
- [Microsoft Internet Explorer settings for Help](#)

Category Graphs and Charts

When building your categories, it is important to take the time to review the category definitions, the documents or records they contain, and how the categories overlap. The visualization pane offers several perspectives on your categories. The Visualization pane is located in the upper right corner of the Categories and Concepts view . If it is not already visible, you can access this pane from the View menu (View > Panes > Visualization).

In this view, the visualization pane offers three perspectives on the commonalities in document or record categorization. The charts and graphs in this pane can be used to analyze your categorization results and aid in fine-tuning categories or reporting. When refining categories, you can use this pane to review your category definitions to uncover categories that are too similar (for example, they share more than 75% of their documents or records) or too distinct. If two categories are too similar, it might help you decide to combine the two categories. Alternatively, you might decide to refine the category definitions by removing certain descriptors from one category or the other.

Depending on what is selected in the Extraction Results pane, Categories pane, or in the Category Definitions dialog box, you can view the corresponding interactions between documents/records and categories on each of the tabs in this pane. Each presents similar information but in a different manner or with a different level of detail. However, in order to refresh a graph for the current selection, click Display on the toolbar of the pane or dialog box in which you have made your selection.

The Visualization pane in the Categories and Concepts view offers the following graphs and charts:

- **Category Bar Chart.** A table and bar chart present the overlap between the documents/records corresponding to your selection and the associated categories. The bar chart also presents ratios of the documents/records in categories to the total number of documents/records. See the topic [Category Bar Chart](#) for more information.
- **Category Web Graph.** This graph presents the document/record overlap for the categories to which the documents/records belong according to the selection in the other panes. See the topic [Category Web Graph](#) for more information.
- **Category Web Table.** This table presents the same information as the Category Web tab but in a table format. The table contains three columns that can be sorted by clicking the column headers. See the topic [Category Web Table](#) for more information.

See the topic [Categorizing text data](#) for more information.

- [Category Bar Chart](#)
- [Category Web Graph](#)
- [Category Web Table](#)
- [Fixing Errors](#)

Related information

- [Visualizing Graphs](#)
- [Cluster Graphs](#)
- [Text Link Analysis Graphs](#)
- [Using Graph Toolbars and Palettes](#)
- [Category Bar Chart](#)
- [Category Web Graph](#)
- [Category Web Table](#)
- [Microsoft Internet Explorer settings for Help](#)

Category Bar Chart

This tab displays a table and bar chart showing the overlap between the documents/records corresponding to your selection and the associated categories. The bar chart also presents ratios of the documents/records in categories to the total number of documents or records . You cannot edit the layout of this chart. You can, however, sort the columns by clicking the column headers.

The chart contains the following columns:

- **Category.** This column presents the name of the categories in your selection. By default, the most common category in your selection is listed first.
- **Bar.** This column presents, in a visual manner, the ratio of the documents or records in a given category to the total number of documents or records.
- **Selection %.** This column presents a percentage based on the ratio of the total number of documents or records for a category to the total number of documents or records represented in the selection.
- **Docs.** This column presents the number of documents or records in a selection for the given category.

Related information

- [Category Graphs and Charts](#)
- [Category Web Graph](#)
- [Category Web Table](#)
- [Microsoft Internet Explorer settings for Help](#)

Category Web Graph

This tab displays a category web graph. The web presents the documents or records overlap for the categories to which the documents or records belong according to the selection in the other panes. If category labels exist, these labels appear in the graph. You can choose a graph layout (network, circle, directed, or grid) using the toolbar buttons in this pane.

In the web, each node represents a category. Using your mouse, you can select and move the nodes within the pane. The size of the node represents the relative size based on the number of documents or records for that category in your selection. The thickness and color of the line between two categories denotes the number of common documents or records they have. If you hover your mouse over a node in Explore mode, a ToolTip displays the name (or label) of the category and the overall number of documents or records in the category.

Note: By default, the Explore mode is enabled for the graphs on which you can move nodes. However, you can switch to Edit mode to edit your graph layouts including colors, fonts, legends, and more. For more information, see [Using Graph Toolbars and Palettes](#).

If you copy the graph data, using the Copy Visualization Data button, and paste it into a spreadsheet or text editor, you will see that the data is given column headers such as V1, V2, through to V7. These columns contain the following information:

- V1, V2 These values correspond to the screen coordinates (X and Y, respectively).
- V3, V5 List the category concept.
- Size, V6 Shows the number of documents the concepts were found in.
- V7 Currently unused.

Related information

- [Category Graphs and Charts](#)
- [Category Bar Chart](#)
- [Category Web Table](#)
- [Microsoft Internet Explorer settings for Help](#)

Category Web Table

This tab displays the same information as the Category Web tab but in a table format. The table contains three columns that can be sorted by clicking the column headers:

- **Count.** This column presents the number of shared, or common, documents or records between the two categories.
- **Category 1.** This column presents the name of the first category followed by the total number of documents or records it contains, shown in parentheses.
- **Category 2.** This column presents the name of the second category followed by the total number of documents or records it contains, shown in parentheses.

Related information

- [Category Graphs and Charts](#)
- [Category Bar Chart](#)
- [Category Web Graph](#)
- [Microsoft Internet Explorer settings for Help](#)

Fixing Errors

Sometimes a visualization cannot be rendered. In this case, the Unable to Render Visualization dialog box appears.

To revert to the state of the visualization before the error occurred, click Undo. Alternatively, you can click Continue to proceed in spite of the error. After continuing, no visualization is displayed until you fix the problem.

Cluster Graphs

After building your clusters, you can explore them visually in the web graphs in the Visualization pane. The visualization pane offers two perspectives on clustering: a Concept Web graph and a Cluster Web graph. The web graphs in this pane can be used to analyze your clustering results and aid in uncovering some concepts and rules you may want to add to your categories. The Visualization pane is located in the upper

right corner of the Clusters view. If it isn't already visible, you can access this pane from the View menu (View > Panes > Visualization). By selecting a cluster in the Clusters pane, you can automatically display the corresponding graphs in the Visualization pane.

Note: By default, the graphs are in the interactive/selection mode in which you can move nodes. However, you can edit your graph layouts in Edit mode, including colors and fonts, legends, and more. See the topic [Using Graph Toolbars and Palettes](#) for more information.

The Clusters view has two web graphs.

- **Concept Web Graph.** This graph presents all of the concepts within the selected cluster(s) as well as linked concepts outside the cluster. This graph can help you see how the concepts within a cluster are linked and any external links. See the topic [Concept Web Graph](#) for more information.
- **Cluster Web Graph.** This graph presents the selected cluster(s) with all of the external links between the selected clusters shown as dotted lines. See the topic [Cluster Web Graph](#) for more information.

See the topic [Analyzing clusters](#) for more information.

- [Concept Web Graph](#)
- [Cluster Web Graph](#)

Related information

- [Visualizing Graphs](#)
- [Category Graphs and Charts](#)
- [Text Link Analysis Graphs](#)
- [Using Graph Toolbars and Palettes](#)
- [Concept Web Graph](#)
- [Cluster Web Graph](#)
- [Microsoft Internet Explorer settings for Help](#)

Concept Web Graph

This tab displays a web graph showing all of the concepts within the selected cluster(s) as well as linked concepts outside the cluster. This graph can help you see how the concepts within a cluster are linked and any external links. Each concept in a cluster is represented as a node, which is color coded according to the type color. See the topic [Creating types](#) for more information.

The internal links between the concepts within a cluster are drawn and the line thickness of each link is directly related to either the doc count for each concept pair's co-occurrence or the similarity link value, depending on your choice on the graph toolbar. The external links between a cluster's concepts and those concepts outside the cluster are also shown.

If concepts are selected in the Cluster Definitions dialog box, the Concept Web graph will display those concepts and any associated internal and external links to those concepts. Any links between other concepts that do not include one of the selected concepts do not appear on the graph.

Note: By default, the graphs are in the interactive/selection mode in which you can move nodes. However, you can edit your graph layouts in Edit mode including colors and fonts, legends, and more. For more information, see [Using Graph Toolbars and Palettes](#).

If you copy the graph data, using the Copy Visualization Data button, and paste it into a spreadsheet or text editor, you will see that the data is given column headers such as V1, V2, through to V7. These columns contain the following information:

- V1, V2 These values correspond to the screen coordinates (X and Y, respectively).
- V3, V6 List the concept type.
- V4, V5 Shows the concept label.
- V7 Currently unused.

Related information

- [Cluster Graphs](#)
- [Cluster Web Graph](#)
- [Microsoft Internet Explorer settings for Help](#)

Cluster Web Graph

This tab displays a web graph showing the selected cluster(s). The external links between the selected clusters as well as any links between other clusters are all shown as dotted lines. In a Cluster Web graph, each node represents an entire cluster and the thickness of lines drawn between them represents the number of external links between two clusters.

Important! In order to display a Cluster Web graph, you must have already built clusters with external links. External links are links between concept pairs in separate clusters (a concept within one cluster and a concept outside in another cluster).

For example, let's say we have two clusters. **Cluster A** has three concepts: **A1**, **A2**, and **A3**. **Cluster B** has two concepts: **B1** and **B2**. The following concepts are linked: **A1-A2**, **A1-A3**, **A2-B1** (External), **A2-B2** (External), **A1-B2** (External), and **B1-B2**. This means that in the Cluster Web graph, the line thickness would represent the three external links.

Note: By default, the graphs are in the interactive/selection mode in which you can move nodes. However, you can edit your graph layouts in Edit mode including colors and fonts, legends, and more. See the topic [Using Graph Toolbars and Palettes](#) for more information.

Related information

- [Cluster Graphs](#)
 - [Concept Web Graph](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Text Link Analysis Graphs

After extracting your Text Link Analysis (TLA) patterns, you can explore them visually in the web graphs in the Visualization pane. The visualization pane offers two perspectives on TLA patterns: a concept (pattern) web graph and a type (pattern) web graph. The web graphs in this pane can be used to visually represent patterns. The Visualization pane is located in the upper right corner of the Text Link Analysis. If it isn't already visible, you can access this pane from the View menu (View > Panes > Visualization). If there is no selection, then the graph area is empty.

Note: By default, the graphs are in the interactive/selection mode in which you can move nodes. However, you can edit your graph layouts in Edit mode including colors and fonts, legends, and more. See the topic [Using Graph Toolbars and Palettes](#) for more information.

The Text Link Analysis view has two web graphs.

- **Concept Web Graph.** This graph presents all the concepts in the selected pattern(s). The line width and node sizes (if type icons are not shown) in a concept graph show the number of global occurrences in the selected table. See the topic [Concept Web Graph](#) for more information.
- **Type Web Graph.** This graph presents all the types in the selected pattern(s). The line width and node sizes (if type icons are not shown) in the graph show the number of global occurrences in the selected table. Nodes are represented by either a type color or by an icon. See the topic [Type Web Graph](#) for more information.

See the topic [Exploring Text Link Analysis](#) for more information.

- [Concept Web Graph](#)
- [Type Web Graph](#)

Related information

- [Exploring Text Link Analysis](#)
 - [Extracting TLA Pattern Results](#)
 - [Type and Concept Patterns](#)
 - [Filtering TLA Results](#)
 - [Data Pane](#)
 - [Concept Web Graph](#)
 - [Type Web Graph](#)
 - [Visualizing Graphs](#)
 - [Category Graphs and Charts](#)
 - [Cluster Graphs](#)
 - [Using Graph Toolbars and Palettes](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Concept Web Graph

This web graph presents all of the concepts represented in the current selection. For example, if you selected a type pattern that had three matching concept patterns, this graph would show three sets of linked concepts. The line width and node sizes in a concept graph represent the global frequency counts. The graph visually represents the same information as what is selected in the patterns panes. The types of each concept are presented either by a color or by an icon depending on what you select on the graph toolbar. See the topic [Using Graph Toolbars and Palettes](#) for more information.

Related information

- [Exploring Text Link Analysis](#)
 - [Extracting TLA Pattern Results](#)
 - [Type and Concept Patterns](#)
 - [Filtering TLA Results](#)
 - [Data Pane](#)
 - [Text Link Analysis Graphs](#)
 - [Type Web Graph](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Type Web Graph

This web graph presents each type pattern for the current selection. For example, if you selected two concept patterns, this graph would show one node per type in the selected patterns and the links between those it found in the same pattern. The line width and node sizes represent the global frequency counts for the set. The graph visually represents the same information as what is selected in the patterns panes. In addition to the type names appearing in the graph, the types are also identified either by their color or by a type icon, depending on what you select on the graph toolbar. See the topic [Using Graph Toolbars and Palettes](#) for more information.

Related information

- [Exploring Text Link Analysis](#)
 - [Extracting TLA Pattern Results](#)
 - [Type and Concept Patterns](#)
 - [Filtering TLA Results](#)
 - [Data Pane](#)
 - [Text Link Analysis Graphs](#)
 - [Concept Web Graph](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Using Graph Toolbars and Palettes

For each graph, there is a toolbar that provides you with quick access to some common palettes from which you can perform a number of actions with your graphs. Each view (Categories and Concepts, Clusters, and Text Link Analysis) has a slightly different toolbar. You can choose between the *Explore* view mode or the *Edit* view mode.

While *Explore* mode allows you to analytically explore the data and values represented by the visualization, *Edit* mode allows you to change the visualization's layout and look. For example, you can change the fonts and colors to match your organization's style guide. To select this mode, choose *View* > *Visualization Pane* > *Edit Mode* from the menus (or click the toolbar icon).

In *Edit* mode, there are several toolbars that affect different aspects of the visualization's layout. If you find that there are any you don't use, you can hide them to increase the amount of space in the dialog box in which the graph is displayed. To select or deselect toolbars, click on the relevant toolbar or palette name on the *View* menu.

Table 1. Text Analytics Toolbar buttons

Button/List	Description
	Enables Edit mode. Switch to the Edit mode to change the look of the graph, such as enlarging the font, changing the colors to match your corporate style guide, or removing labels and legends.
	Enables Explore mode. By default, the Explore mode is turned on, which means that you can move and drag nodes around the graph as well as hover over graph objects to reveal additional ToolTip information.
	Select a type of web display for the graphs in the Categories and Concepts view as well as the Text Link Analysis view. <ul style="list-style-type: none">• Circle Layout A general layout that can be applied to any graph. It lays out a graph assuming that links are undirected and treats all nodes the same. Nodes are only placed around the perimeter of a circle.• Network Layout A general layout that can be applied to any graph. It lays out a graph assuming that links are undirected and treats all nodes the same. Nodes are placed freely within the layout.• Directed Layout A layout that should only be used for directed graphs. This layout produces treelike structures from root nodes down to leaf nodes and organizes by colors. Hierarchical data tends to display nicely with this layout.• Grid Layout A general layout that can be applied to any graph. It lays out a graph assuming that links are undirected and treats all nodes the same. Nodes are only placed at grid points within the space.

Button/List	Description
	Link size representation. Choose what the thickness of the line represents in the graph. This only applies to the Clusters view. The Clusters web graph only shows the number of external links between clusters. You can choose between: <ul style="list-style-type: none"> Similarity Thickness indicates the number of external links between two clusters Co-occurrence Thickness indicates the number of documents in which a co-occurrence of descriptors takes place.
	A toggle button that, when pressed, displays the legend. When the button is not pushed, the legend is not shown.
	A toggle button that, when pressed, displays the type icons in the graph rather than type colors. This only applies to Text Link Analysis view.
	A toggle button that, when pressed, displays the Links Slider beneath the graph. You can filter the results by sliding the arrow.
	Will display the graph for highest level of categories selected rather than for their subcategories.
	Will display the graph for lowest level of categories selected.
	This option controls how the names of subcategories are displayed in the output. <ul style="list-style-type: none"> Full category path This option will output the name of the category and the full path of parent categories if applicable using slashes to separate category names from subcategory names. Short category path This option will output only the name of the category but use ellipses to show the number of parent categories for the category in question. Bottom level category This option will output only the name of the category without the full path or parent categories shown.

Related information

- [Visualizing Graphs](#)
- [Category Graphs and Charts](#)
- [Cluster Graphs](#)
- [Text Link Analysis Graphs](#)
- [Microsoft Internet Explorer settings for Help](#)

Session Resource Editor

IBM® SPSS® Modeler Text Analytics rapidly and accurately captures and extracts key concepts from text data. This extraction process relies heavily on linguistic resources to dictate how to extract information from text data. By default, these resources come from resource templates.

IBM SPSS Modeler Text Analytics is shipped with a set of specialized **resource templates** that contain a set of linguistic and nonlinguistic resources, in the form of libraries and advanced resources, to help define how your data will be handled and extracted. See the topic [Templates and Resources](#) for more information.

In the node dialog box, you can load a copy of the template's resources into the node. Once inside an interactive workbench session, you can customize these resources specifically for this node's data, if you wish. During an interactive workbench session, you can work with your resources in the Resource Editor view. Whenever an interactive session is launched, an extraction is performed using the resources loaded in the node dialog box, unless you have cached your data and extraction results in your node.

- [Editing resources in the Resource Editor](#)
- [Making and Updating Templates](#)
- [Switching resource templates](#)

Related information

- [Making and Updating Templates](#)
- [Switching resource templates](#)
- [Templates and Resources](#)
- [Template Editor vs. Resource Editor](#)
- [The Editor interface](#)
- [Opening Templates](#)
- [Saving Templates](#)
- [Updating Node Resources After Loading](#)
- [Managing Templates](#)
- [Importing and Exporting Templates](#)
- [Backing Up Resources](#)
- [Importing resource files](#)
- [Editing resources in the Resource Editor](#)

- [Microsoft Internet Explorer settings for Help](#)
- [Working with Libraries](#)
- [About Library Dictionaries](#)
- [About Advanced Resources](#)
- [About Text Link Rules](#)

Editing resources in the Resource Editor

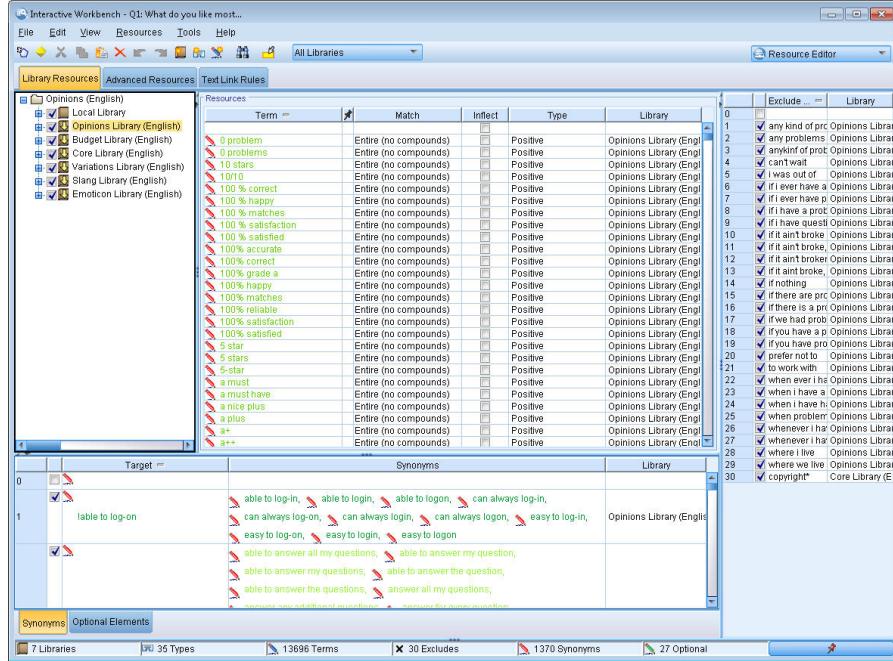
The Resource Editor offers access to the set of resources used to produce the extraction results (concepts, types, and patterns) for an interactive workbench session. This editor is very similar to the Template Editor except that in the Resource Editor you are editing the resources for this session. When you are finished working on your resources and any other work you've done, you can update the modeling node to save this work so that it can be restored in a subsequent interactive workbench session. See the topic [Updating Modeling Nodes and Saving](#) for more information.

If you want to work directly on the templates used to load resources into nodes, we recommend you use the Template Editor. Many of the tasks you can perform inside the Resource Editor are performed just like they are in the Template Editor, such as:

- Working with libraries. See the topic [Working with Libraries](#) for more information.
- Creating type dictionaries. See the topic [Creating types](#) for more information.
- Adding terms to dictionaries. See the topic [Adding terms](#) for more information.
- Creating synonyms. See the topic [Defining synonyms](#) for more information.
- Importing and exporting templates. See the topic [Importing and Exporting Templates](#) for more information.
- Publishing libraries. See the topic [Publishing Libraries](#) for more information.

For Dutch, English, French, German, Italian, Portuguese, and Spanish Text

Figure 1. Resource Editor view



Making and Updating Templates

Whenever you make changes to your resources and want to reuse them in the future, you can save the resources as a template. When doing so, you can choose to save using an existing template name or by providing a new name. Then, whenever you load this template in the future, you'll be able to obtain the same resources. See the topic [Copying resources from templates and TAPs](#) for more information.

Note: You can also publish and share your libraries. See the topic [Sharing Libraries](#) for more information.

To Make (or Update) a Template

1. From the menus in the Resource Editor view, choose Resources > Make Resource Template. The Make Resource Template dialog box opens.

2. Enter a new name in the Template Name field, if you want to make a new template. Select a template in the table, if you want to overwrite an existing template with the currently loaded resources.
3. Click Save to make the template.

Important! Since templates are loaded when you select them in the node and not when the stream is executed, please make sure to reload the resource template in any other nodes in which it is used if you want to get the latest changes. See the topic [Updating Node Resources After Loading](#) for more information.

Related information

- [Editing resources in the Resource Editor](#)
- [Switching resource templates](#)
- [Updating Node Resources After Loading](#)
- [Managing Templates](#)
- [Importing and Exporting Templates](#)
- [Backing Up Resources](#)
- [Importing resource files](#)
- [Session Resource Editor](#)
- [Templates and Resources](#)
- [Template Editor vs. Resource Editor](#)
- [The Editor interface](#)
- [Opening Templates](#)
- [Saving Templates](#)
- [Microsoft Internet Explorer settings for Help](#)

Switching resource templates

If you want to replace the resources currently loaded in the session with a copy of those from another template, you can switch to those resources. Doing so will overwrite any resources currently loaded in the session. If you are switching resources in order to have some predefined Text Link Analysis (TLA) pattern rules, make sure to select a template that has them marked in the TLA column.

Switching resources is particularly useful when you want to restore the session work (categories, patterns, and resources) but want to load an updated copy of the resources from a template without losing your other session work. You can select the template whose contents you want to copy into the Resource Editor and click OK. This replaces the resources you have in this session. Make sure you update the modeling node at the end of your session if you want to keep these changes next time you launch the interactive workbench session.

Note: If you switch to the contents of another template during an interactive session, the name of the template listed in the node will still be the name of the last template loaded and copied. In order to benefit from these resources or other session work, update your modeling node before exiting the session and select the Use session work option in the node. See the topic [Updating Modeling Nodes and Saving](#) for more information.

To switch resources

1. From the menus in the Resource Editor view, choose Resources > Switch Resource Templates. The Switch Resources dialog box opens.
2. Select the template you want to use from those shown in the table.
3. Click OK to abandon those resources currently loaded and load a copy of those in the selected template in their place. If you have made changes to your resources and want to save your libraries for a future use, you can publish, update, and share them before switching. See the topic [Sharing Libraries](#) for more information.

Templates and Resources

IBM® SPSS® Modeler Text Analytics rapidly and accurately captures and extracts key concepts from text data. This extraction process relies heavily on linguistic resources to dictate how to extract information from text data. See the topic [How extraction works](#) for more information. You can fine-tune these resources in the Resource Editor view.

When you install the software, you also get a set of specialized resources. These shipped resources allow you to benefit from years of research and fine-tuning for specific languages and specific applications. Since the shipped resources may not always be perfectly adapted to the context of your data, you can edit these resource templates or even create and use custom libraries uniquely fine-tuned to your organization's data. These resources come in various forms and each can be used in your session. Resources can be found in the following:

- **Resource templates.** Templates are made up of a set of libraries, types, and some advanced resources which together form a specialized set of resources adapted to a particular domain or context such as product opinions.
- **Text analysis packages (TAP).** In addition to the resources stored in a template, TAPs also bundle together one or more specialized category sets generated using those resources so that both the categories and the resources are stored together and reusable. See the

topic [Using Text Analysis Packages](#) for more information.

- **Libraries.** Libraries are used as building blocks for both TAPs and templates. They can also be added individually to resources in your session. Each library is made up of several dictionaries used to define and manage types, synonyms, and exclude lists. While libraries are also delivered individually, they are prepackaged together in templates and TAPs. See the topic [Working with Libraries](#) for more information.

Note: During extraction, some compiled internal resources are also used. These compiled resources contain a large number of definitions complementing the types in the Core library. These compiled resources cannot be edited.

The Resource Editor offers access to the set of resources used to produce the extraction results (concepts, types, and patterns). There are a number of tasks you might perform in the Resource Editor and they include:

- **Working with libraries.** See the topic [Working with Libraries](#) for more information.
 - **Creating type dictionaries.** See the topic [Creating types](#) for more information.
 - **Adding terms to dictionaries.** See the topic [Adding terms](#) for more information.
 - **Creating synonyms.** See the topic [Defining synonyms](#) for more information.
 - **Updating the resources in TAPs.** See the topic [Updating Text Analysis Packages](#) for more information.
 - **Making templates.** See the topic [Making and Updating Templates](#) for more information.
 - **Importing and exporting templates.** See the topic [Importing and Exporting Templates](#) for more information.
 - **Publishing libraries.** See the topic [Publishing Libraries](#) for more information.
-
- [Template Editor vs. Resource Editor](#)
 - [The Editor interface](#)
 - [Opening Templates](#)
 - [Saving Templates](#)
 - [Updating Node Resources After Loading](#)
 - [Managing Templates](#)
 - [Importing and Exporting Templates](#)
 - [Exiting the Template Editor](#)
 - [Backing Up Resources](#)
 - [Importing resource files](#)

Related information

- [Session Resource Editor](#)
- [Making and Updating Templates](#)
- [Switching resource templates](#)
- [Template Editor vs. Resource Editor](#)
- [The Editor interface](#)
- [Opening Templates](#)
- [Saving Templates](#)
- [Updating Node Resources After Loading](#)
- [Managing Templates](#)
- [Importing and Exporting Templates](#)
- [Backing Up Resources](#)
- [Importing resource files](#)
- [Editing resources in the Resource Editor](#)
- [Microsoft Internet Explorer settings for Help](#)
- [Working with Libraries](#)
- [About Library Dictionaries](#)
- [About Advanced Resources](#)
- [About Text Link Rules](#)

Template Editor vs. Resource Editor

There are two main methods for working with and editing your templates, libraries, and their resources. You can work on linguistic resources in the Template Editor or the Resource Editor.

Template Editor

The Template Editor allows you to create and edit resource templates without an interactive workbench session and independent of a specific node or stream. You can use this editor to create or edit resource templates before loading them into the Text Link Analysis node and the Text Mining modeling node.

The Template Editor is accessible through the main IBM® SPSS® Modeler toolbar from the Tools > Text Analytics Template Editor menu.

Resource Editor

The Resource Editor, which is accessible within an interactive workbench session, allows you to work with the resources in the context of a specific node and dataset. When you add a Text Mining modeling node to a stream, you can load a copy of a resource template's content or a copy of a text analysis package (category sets and resources) to control how text is extracted for text mining. When you launch an interactive workbench session, in addition to creating categories, extracting text link analysis patterns, and creating category models, you can also fine-tune the resources for that session's data in the integrated Resource Editor view. See the topic [Editing resources in the Resource Editor](#) for more information.

Whenever you work on the resources in an interactive workbench session, those changes apply only to that session. If you want to save your work (resources, categories, patterns, etc.) so you can continue in a subsequent session, you must update the modeling node. See the topic [Updating Modeling Nodes and Saving](#) for more information.

If you want to save your changes back to the original template, whose contents were copied into the modeling node, so that this updated template can be loaded into other nodes, you can make a template from the resources. See the topic [Making and Updating Templates](#) for more information.

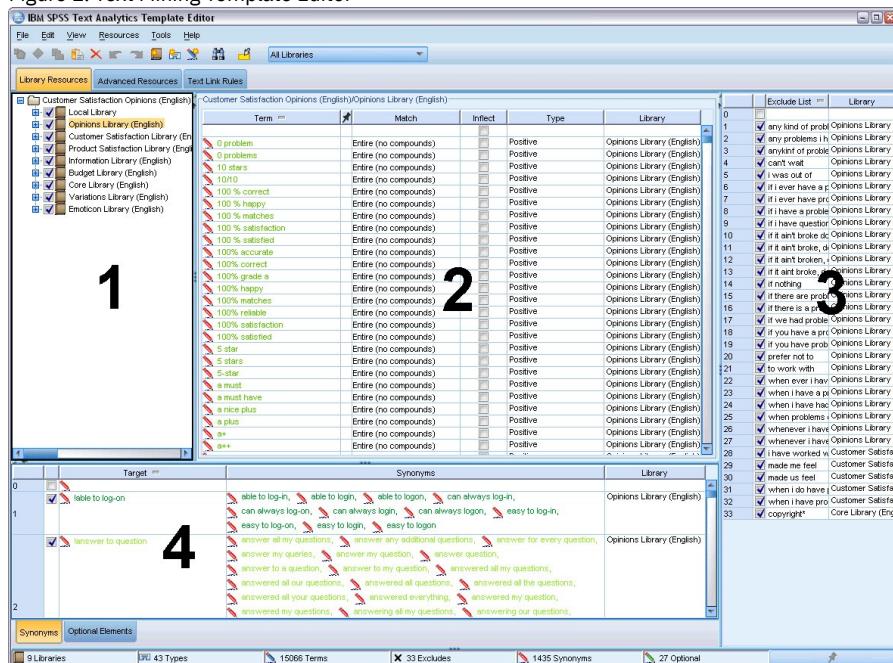
Note: If you make changes to templates or libraries and save them to a backup directory, and then upgrade your version of IBM SPSS Modeler Text Analytics, you will be given the option of importing your custom templates and libraries. The first time you run a SPSS Modeler Text Analytics stream or open the Resource Editor after an upgrade, default templates and libraries are copied to your machine. A Saved templates warning or a Saved libraries warning (or both) is displayed with a list of the templates and/or libraries that are updated as part of the product upgrade, and you are given the option of importing your custom templates and libraries from the directory in which you saved them. After clicking OK in the warning message, you can open the Manage resource templates dialog or the Manage libraries dialog at any time to choose which custom templates or libraries you want to import.

The Editor interface

The operations that you perform in the Template Editor or Resource Editor revolve around the management and fine-tuning of the linguistic resources. These resources are stored in the form of templates and libraries. See the topic [Type dictionaries](#) for more information.

Library Resources tab

Figure 1. Text Mining Template Editor



The interface is organized into four parts, as follows:

1. Library Tree pane. Located in the upper left corner, this pane displays a tree of the libraries. You can enable and disable libraries in this tree as well as filter the views in the other panes by selecting a library in the tree. You can perform many operations in this tree using the context menus. If you expand a library in the tree, you can see the set of types it contains. You can also filter this list through the View menu if you want to focus on a particular library only.

2. Term Lists from Type Dictionaries pane. Located to the right of the library tree, this pane displays the term lists of the type dictionaries for the libraries selected in the tree. A **type dictionary** is a collection of terms to be grouped under one label, or type, name. When the extraction engine reads your text data, it compares words found in the text to the terms in the type dictionaries. If an extracted concept appears as a term

in a type dictionary, then that type name is assigned. You can think of the type dictionary as a distinct dictionary of terms that have something in common. For example, the `<Location>` type in the Core library contains concepts such as `new orleans`, `great britain`, and `new york`. These terms all represent geographical locations. A library can contain one or more type dictionaries. See the topic [Type dictionaries](#) for more information.

3. Exclude Dictionary pane. Located on the right side, this pane displays the collection of terms that will be excluded from the final extraction results. The terms appearing in this exclude dictionary do not appear in the Extraction Results pane. Excluded terms can be stored in the library of your choosing. However, the Exclude Dictionary pane displays all of the excluded terms for all libraries visible in the library tree. See the topic [Exclude dictionaries](#) for more information.

4. Substitution Dictionary pane. Located in the lower left, this pane displays synonyms and optional elements, each in their own tab. Synonyms and optional elements help group similar terms under one lead, or target, concept in the final extraction results. This dictionary can contain known synonyms and user-defined synonyms and elements, as well as common misspellings paired with the correct spelling. Synonym definitions and optional elements can be stored in the library of your choosing. However, the substitution dictionary pane displays all of the contents for all libraries visible in the library tree. While this pane displays all synonyms or optional elements from all libraries, the substitutions for all of the libraries in the tree are shown together in this pane. A library can contain only one substitution dictionary. See the topic [Substitution/Synonym dictionaries](#) for more information.

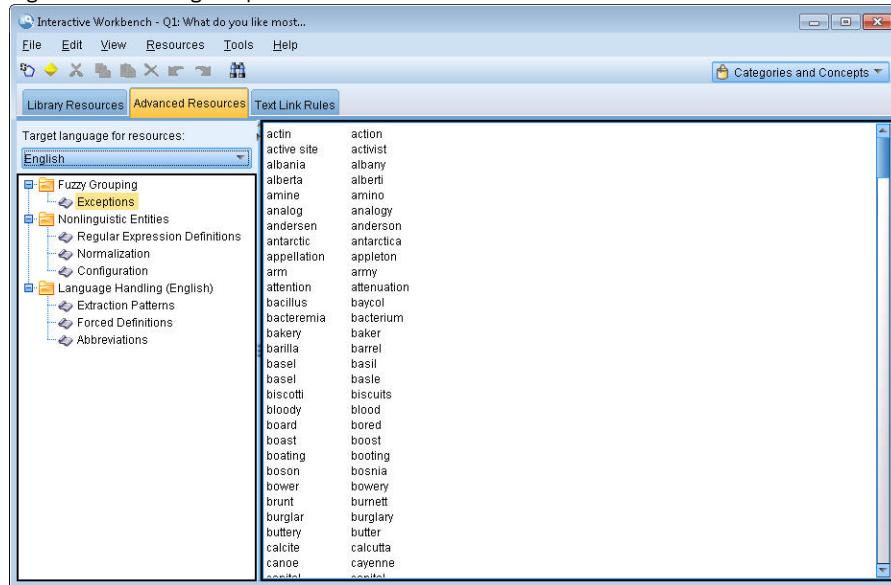
Notes:

- If you want to filter so that you see only the information pertaining to a single library, you can change the library view using the drop-down list on the toolbar. It contains a top-level entry called All Libraries as well as an additional entry for each individual library. See the topic [Viewing Libraries](#) for more information.

Advanced Resources tab

The advanced resources are available from the second tab of the editor view. You can review and edit the advanced resources in this tab. See the topic [About Advanced Resources](#) for more information.

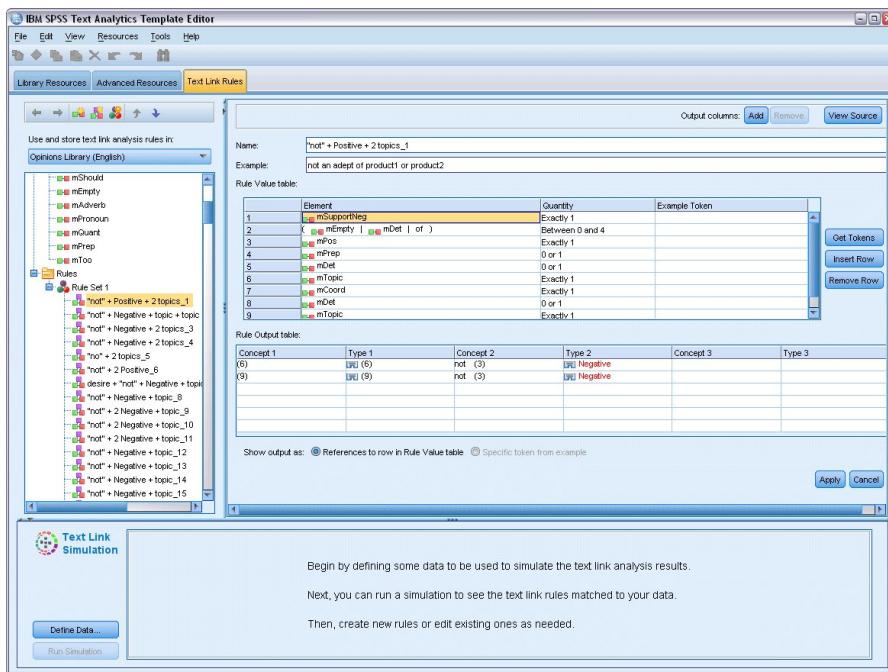
Figure 2. Text Mining Template Editor - Advanced Resources tab



Text Link Rules tab

Since version 14, the text link analysis rules are editable in their own tab of the editor view. You can work in the rule editor, create your own rules, and even run simulations to see how your rules impact the TLA results. See the topic [About Text Link Rules](#) for more information.

Figure 3. Text Mining Template Editor - Text Link Rules tab



Opening Templates

When you launch the Template Editor, you are prompted to open a template. Likewise, you can open a template from the File menu. If you want a template that contains some Text Link Analysis (TLA) rules, make sure to select a template that has an icon in the TLA column. The language for which a template was created is shown in the Language column.

If you want to import a template that isn't shown in the table or if you want to export a template, you can use the buttons in the Open Template dialog box. See the topic [Importing and Exporting Templates](#) for more information.

To Open a Template

1. From the menus in the Template Editor, choose File > Open Resource Template. The Open Resource Template dialog box opens.
2. Select the template you want to use from those shown in the table.
3. Click OK to open this template. If you currently have another template open in the editor, clicking OK will abandon that template and display the template you selected here. If you have made changes to your resources and want to save your libraries for a future use, you can publish, update, and share them before opening another. See the topic [Sharing Libraries](#) for more information.

Related information

- [Session Resource Editor](#)
- [Making and Updating Templates](#)
- [Switching resource templates](#)
- [Templates and Resources](#)
- [Template Editor vs. Resource Editor](#)
- [The Editor interface](#)
- [Saving Templates](#)
- [Updating Node Resources After Loading](#)
- [Managing Templates](#)
- [Importing and Exporting Templates](#)
- [Backing Up Resources](#)
- [Importing resource files](#)
- [Editing resources in the Resource Editor](#)
- [Microsoft Internet Explorer settings for Help](#)

Saving Templates

In the Template Editor, you can save the changes you made to a template. You can choose to save using an existing template name or by providing a new name.

If you make changes to a template that you've already loaded into a node at a previous time, you will have to reload the template contents into the node to get the latest changes. See the topic [Copying resources from templates and TAPs](#) for more information.

Or, if you are using the option Use saved interactive work in the Model tab of the Text Mining node, meaning you are using resources from a previous interactive workbench session, you'll need to switch to this template's resources from within the interactive workbench session. See the topic [Switching resource templates](#) for more information.

Note: You can also publish and share your libraries. See the topic [Sharing Libraries](#) for more information.

To Save a Template

1. From the menus in the Template Editor, choose File > Save Resource Template. The Save Resource Template dialog box opens.
2. Enter a new name in the Template name field, if you want to save this template as a new template. Select a template in the table, if you want to overwrite an existing template with the currently loaded resources.
3. If desired, enter a description to display a comment or annotation in the table.
4. Click Save to save the template.

Important! Since resources from templates or TAPs are loaded/copied into the node, you must update the resources by reloading them if you make changes to a template and want to benefit from these changes in an existing stream. See the topic [Updating Node Resources After Loading](#) for more information.

Related information

- [Session Resource Editor](#)
- [Making and Updating Templates](#)
- [Switching resource templates](#)
- [Templates and Resources](#)
- [Template Editor vs. Resource Editor](#)
- [The Editor interface](#)
- [Opening Templates](#)
- [Updating Node Resources After Loading](#)
- [Managing Templates](#)
- [Importing and Exporting Templates](#)
- [Backing Up Resources](#)
- [Importing resource files](#)
- [Editing resources in the Resource Editor](#)
- [Microsoft Internet Explorer settings for Help](#)

Updating Node Resources After Loading

By default, when you add a node to a stream, a set of resources from a default template are loaded and embedded into your node. And if you change templates or use a TAP, when you load them, a copy of those resources then overwrites the resources. Since templates and TAPs are not linked to the node directly, any changes you make to a template or TAP are not automatically available in a preexisting node. In order to benefit from those changes, you would have to update the resources in that node. The resources can be updated in one of two ways.

Method 1: Reloading Resources in Model Tab

If you want to update the resources in the node using a new or updated template or TAP, you can reload it in the Model tab of the node. By reloading, you will replace the copy of the resources in the node with a more current copy. For your convenience, the updated time and date will appear on the Model tab along with the originating template's name. See the topic [Copying resources from templates and TAPs](#) for more information.

However, if you are working with interactive session data in a Text Mining modeling node and you have selected the Use session work option on the Model tab, the saved session work and resources will be used and the Load button is disabled. It is disabled because, at one time during an interactive workbench session, you chose the Update Modeling Node option and kept the categories, resources, and other session work. In that case, if you want to change or update those resources, you can try the next method of switching the resources in the Resource Editor.

Method 2: Switching Resources in the Resource Editor

Anytime you want to use different resources during an interactive session, you can exchange those resources using the Switch Resources dialog box. This is especially useful when you want to reuse existing category work but replace the resources. In this case, you can select the Use session work option on the Model tab of a Text Mining modeling node. Doing so will disable the ability to reload a template through the node dialog box and instead keep the settings and changes you made during your session. Then you can launch the interactive workbench session by executing the stream and switch the resources in the Resource Editor. See the topic [Switching resource templates](#) for more information.

In order to keep session work for subsequent sessions, including the resources, you need to update the modeling node from within the interactive workbench session so that the resources (and other data) are saved back to the node. See the topic [Updating Modeling Nodes and Saving](#) for more information.

Note: If you switch to the contents of another template during an interactive session, the name of the template listed in the node will still be the name of the last template loaded and copied. In order to benefit from these resources or other session work, update your modeling node before exiting the session.

Related information

- [Session Resource Editor](#)
 - [Making and Updating Templates](#)
 - [Switching resource templates](#)
 - [Templates and Resources](#)
 - [Template Editor vs. Resource Editor](#)
 - [The Editor interface](#)
 - [Opening Templates](#)
 - [Saving Templates](#)
 - [Managing Templates](#)
 - [Importing and Exporting Templates](#)
 - [Backing Up Resources](#)
 - [Importing resource files](#)
 - [Editing resources in the Resource Editor](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Managing Templates

There are also some basic management tasks you might want to perform from time to time on your templates, such as renaming your templates, importing and exporting templates, or deleting obsolete templates. These tasks are performed in the Manage Templates dialog box. Importing and exporting templates enables you to share templates with other users. See the topic [Importing and Exporting Templates](#) for more information.

Note: You cannot rename or delete the templates that are installed (or shipped) with this product. Instead, if you want to rename, you can open the installed template and make a new one with the name of your choice. You can delete your custom templates; however, if you try to delete a shipped template, it will be reset to the version originally installed.

To Rename a Template

1. From the menus, choose Resources > Manage Resource Templates. The Manage Templates dialog box opens.
2. Select the template you want to rename and click Rename. The name box becomes an editable field in the table.
3. Type a new name and press the Enter key. A confirmation dialog box opens.
4. If you are satisfied with the name change, click Yes. If not, click No.

To Delete a Template

1. From the menus, choose Resources > Manage Resource Templates. The Manage Templates dialog box opens.
2. In the Manage Templates dialog box, select the template you want to delete.
3. Click Delete. A confirmation dialog box opens.
4. Click Yes to delete or click No to cancel the request. If you click Yes, the template is deleted.

Related information

- [Editing resources in the Resource Editor](#)
- [Making and Updating Templates](#)
- [Switching resource templates](#)
- [Updating Node Resources After Loading](#)
- [Importing and Exporting Templates](#)
- [Backing Up Resources](#)
- [Importing resource files](#)
- [Exiting the Template Editor](#)
- [Session Resource Editor](#)
- [Templates and Resources](#)
- [Template Editor vs. Resource Editor](#)
- [The Editor interface](#)
- [Opening Templates](#)
- [Saving Templates](#)
- [Microsoft Internet Explorer settings for Help](#)

Importing and Exporting Templates

You can share templates with other users or machines by importing and exporting them. Templates are stored in an internal database but can be exported as *.lrt files to your hard drive.

Since there are circumstances under which you might want to import or export templates, there are several dialog boxes that offer those capabilities.

- Open Template dialog box in the Template Editor
- Load Resources dialog box in the Text Mining modeling node and Text Link Analysis node.
- Manage Templates dialog box in the Template Editor and the Resource Editor.

To Import a Template

1. In the dialog box, click Import. The Import Template dialog box opens.
2. Select the resource template file (*.lrt) to import and click Import. You can save the template you are importing with another name or overwrite the existing one. The dialog box closes, and the template now appears in the table.

To Export a Template

1. In the dialog box, select the template you want export and click Export. The Select Directory dialog box opens.
2. Select the directory to which you want to export and click Export. This dialog box closes, and the template is exported and carries the file extension (*.lrt)

Related information

- [Managing Templates](#)
- [Exiting the Template Editor](#)
- [Backing Up Resources](#)
- [Importing resource files](#)
- [Session Resource Editor](#)
- [Making and Updating Templates](#)
- [Switching resource templates](#)
- [Templates and Resources](#)
- [Template Editor vs. Resource Editor](#)
- [The Editor interface](#)
- [Opening Templates](#)
- [Saving Templates](#)
- [Updating Node Resources After Loading](#)
- [Editing resources in the Resource Editor](#)
- [Microsoft Internet Explorer settings for Help](#)

Exiting the Template Editor

When you are finished working in the Template Editor, you can save your work and exit the editor.

To Exit the Template Editor

1. From the menus, choose File > Close. The Save and Close dialog box opens.
2. Select Save changes to template in order to save the open template before closing the editor.
3. Select Publish libraries if you want to publish any of the libraries in the open template before closing the editor. If you select this option, you will be prompted to select the libraries to publish. See the topic [Publishing Libraries](#) for more information.

Related information

- [Managing Templates](#)
- [Importing and Exporting Templates](#)
- [Backing Up Resources](#)
- [Importing resource files](#)
- [Microsoft Internet Explorer settings for Help](#)

Backing Up Resources

You may want to back up your resources from time to time as a security measure.

Important! When you restore, the entire contents of your resources will be wiped clean and only the contents of the backup file will be accessible in the product. This includes any open work.

Note: You can only backup and restore to the same major version of your software. For example, if you backup from version 15, you cannot restore that backup to version 16.

To Back Up the Resources

1. From the menus, choose Resources > Backup Tools > Backup Resources. The Backup dialog box opens.
2. Enter a name for your backup file and click Save. The dialog box closes, and the backup file is created.

To Restore the Resources

1. From the menus, choose Resources > Backup Tools > Restore Resources. An alert warns you that restoring will overwrite the current contents of your database.
2. Click Yes to proceed. The dialog box opens.
3. Select the backup file you want to restore and click Open. The dialog box closes, and resources are restored in the application.

Related information

- [Managing Templates](#)
 - [Importing and Exporting Templates](#)
 - [Exiting the Template Editor](#)
 - [Importing resource files](#)
 - [Session Resource Editor](#)
 - [Making and Updating Templates](#)
 - [Switching resource templates](#)
 - [Templates and Resources](#)
 - [Template Editor vs. Resource Editor](#)
 - [The Editor interface](#)
 - [Opening Templates](#)
 - [Saving Templates](#)
 - [Updating Node Resources After Loading](#)
 - [Editing resources in the Resource Editor](#)
 - [Working with Libraries](#)
 - [Shipped libraries](#)
 - [Creating Libraries](#)
 - [Adding public libraries](#)
 - [Finding Terms and Types](#)
 - [Viewing Libraries](#)
 - [Managing Local Libraries](#)
 - [Managing Public Libraries](#)
 - [Sharing Libraries](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Importing resource files

If you have made changes directly in resource files outside of this product, you can import them into a selected library by selecting that library and proceeding with the import. When you import a directory, you can import all of supported files into a specific open library as well. You can only import *.txt files.

Each imported file must contain only one entry per line, and if the contents are structured as:

- A list words or phrases (one per line). The file is imported as a term list for a type dictionary, where the type dictionary takes the name of the file minus the extension.
- A list of entries such as `term1 <TAB> term2`, then it is imported as a list of synonyms, where `term1` is the set of the underlying term and `term2` is the target term.

To import a single resource file

1. From the menus, choose Resources > Import Files > Import Single File. The Import File dialog box opens.
2. Select the file you want to import and click Import. The file contents are transformed into an internal format and added to your library.

To import all files in a directory

1. From the menus, choose Resources > Import Files > Import Entire Directory. The Import Directory dialog box opens.
 2. Select the library in which you want all of the resource files imported from the Import list. If you select the Default option, a new library will be created using the name of the directory as its name.
 3. Select the directory from which to import the files. Subdirectories will not be read.
 4. Click Import. The dialog box closes and the content from those imported resource files now appears in the editor in the form of dictionaries and advanced resource files.
-

Working with Libraries

The resources used by the extraction engine to extract and group terms from your text data always contain one or more libraries. You can see the set of libraries in the library tree located in the upper left part of the Template Editor and Resource Editor. The libraries are composed of three kinds of dictionaries:

- Type dictionaries
- Substitution dictionaries
- Exclude dictionaries

See the topic [About Library Dictionaries](#) for more information.

The resource template or the resources from the TAP you chose includes several libraries to enable you to immediately begin extracting concepts from your text data. However, you can create your own libraries as well and also publish them so you can reuse them. See the topic [Publishing Libraries](#) for more information.

For example, suppose that you frequently work with text data related to the automotive industry. After analyzing your data, you decide that you would like to create some customized resources to handle industry-specific vocabulary or jargon. Using the Template Editor , you can create a new template, and in it a library to extract and group automotive terms. Since you will need the information in this library again, you publish your library to a central repository, accessible in the **Manage Libraries** dialog box, so that it can be reused independently in different stream sessions .

Suppose that you are also interested in grouping terms that are specific to different subindustries, such as electronic devices, engines, cooling systems, or even a particular manufacturer or market. You can create a library for each group and then publish the libraries so that they can be used with multiple sets of text data. In this way, you can add the libraries that best correspond to the context of your text data.

Note: Additional resources can be configured and managed in the Advanced Resources tab. Some apply to all of the libraries and manage nonlinguistic entities, fuzzy grouping exceptions, and so on. Additionally, you can edit the text link analysis pattern rules, which are library-specific, in the Text Link Rules tab as well. See the topic [About Advanced Resources](#) for more information.

- [Shipped Libraries](#)
- [Creating Libraries](#)
- [Adding public libraries](#)
- [Finding Terms and Types](#)
- [Viewing Libraries](#)
- [Managing Local Libraries](#)
- [Managing Public Libraries](#)
- [Sharing Libraries](#)
- [Resolving Conflicts](#)

Related information

- [Backing Up Resources](#)
- [Importing resource files](#)
- [Shipped libraries](#)
- [Creating Libraries](#)
- [Adding public libraries](#)
- [Finding Terms and Types](#)
- [Viewing Libraries](#)
- [Managing Local Libraries](#)
- [Managing Public Libraries](#)
- [Sharing Libraries](#)
- [Microsoft Internet Explorer settings for Help](#)
- [Session Resource Editor](#)
- [Templates and Resources](#)
- [About Library Dictionaries](#)
- [About Advanced Resources](#)
- [About Text Link Rules](#)

Shipped libraries

By default, several libraries are installed with IBM® SPSS® Modeler Text Analytics. You can use these preformatted libraries to access thousands of predefined terms and synonyms as well as many different types. These shipped libraries are fine-tuned to several different domains and are available in several different languages.

There are a number of libraries but the most commonly used are as follows:

- Local library. Used to store user-defined dictionaries. It is an empty library added by default to all resources. It contains an empty type dictionary too. It is most useful when making changes or refinements to the resources directly (such as adding a word to a type) from the Categories and Concepts view, Clusters view, and the Text Link Analysis view . In this case, those changes and refinements are automatically stored in the first library listed in the library tree in the Resource Editor; by default, this is the *Local Library*. You cannot publish this library because it is specific to the session data. If you want to publish its contents, you must rename the library first.
- Core library. Used in most cases, since it comprises the basic five built-in types representing people, locations, organizations, products, and unknown. While you may see only a few terms listed in one of its type dictionaries, the types represented in the Core library are actually complements to the robust types found in the internal, compiled resources delivered with your text-mining product. These internal, compiled resources contain thousands of terms for each type. For this reason, while you may not see a term in the type dictionary term list, it can still be extracted and typed with a Core type. This explains how names such as *George* can be extracted and typed as *<Person>* when only *John* appears in the *<Person>* type dictionary in the Core library. Similarly, if you do not include the Core library, you may still see these types in your extraction results, since the compiled resources containing these types will still be used by the extraction engine.
- Opinions library. Used most commonly to extract opinions and sentiments from text data. This library includes thousands of words representing attitudes, qualifiers, and preferences that—when used in conjunction with other terms—indicate an opinion about a subject. This library includes a number of built-in types, synonyms, and excludes. It also includes a large set of pattern rules used for text link analysis. To benefit from the text link analysis rules in this library and the pattern results they produce, this library must be specified in the Text Link Rules tab. See the topic [About Text Link Rules](#) for more information.
- Budget library. Used to extract terms referring to the cost of something. This library includes many words and phrases that represent adjectives, qualifiers, and judgments regarding the price or quality of something.
- Variations library. Used to include cases where certain language variations require synonym definitions to properly group them. This library includes only synonym definitions.

Although some of the libraries shipped outside the templates resemble the contents in some templates, the templates have been specifically tuned to particular applications and contain additional advanced resources. We recommend that you try to use a template that was designed for the kind of text data you are working with and make your changes to those resources rather than just adding individual libraries to a more generic template.

Compiled resources are also delivered with IBM SPSS Modeler Text Analytics. They are always used during the extraction process and contain a large number of complementary definitions to the built-in type dictionaries in the default libraries. Since these resources are compiled, they cannot be viewed or edited. You can, however, force a term that was typed by these compiled resources into any other dictionary. See the topic [Forcing terms](#) for more information.

Creating Libraries

You can create any number of libraries. After creating a new library, you can begin to create type dictionaries in this library and enter terms, synonyms, and excludes.

To Create a Library

1. From the menus, choose Resources > New Library. The Library Properties dialog opens.
2. Enter a name for the library in the Name text box.
3. If desired, enter a comment in the Annotation text box.
4. Click Publish if you want to publish this library now before entering anything in the library. See the topic [Sharing Libraries](#) for more information. You can also publish later at any time.
5. Click OK to create the library. The dialog box closes and the library appears in the tree view. If you expand the libraries in the tree, you will see that an empty type dictionary has been automatically included in the library. In it, you can immediately begin adding terms. See the topic [Adding terms](#) for more information.

Related information

- [Backing Up Resources](#)
- [Importing resource files](#)
- [Working with Libraries](#)
- [Shipped libraries](#)
- [Adding public libraries](#)

- [Finding Terms and Types](#)
 - [Viewing Libraries](#)
 - [Managing Local Libraries](#)
 - [Managing Public Libraries](#)
 - [Sharing Libraries](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Adding public libraries

If you want to reuse a library from another session data, you can add it to your current resources as long as it is a public library. A *public library* is a library that has been published. See the topic [Publishing Libraries](#) for more information.

When you add a public library, a *local* copy is embedded into your session data. You can make changes to this library; however, you must republish the public version of the library if you want to share the changes.

When adding a public library, a Resolve Conflicts dialog box may appear if any conflicts are discovered between the terms and types in one library and the other local libraries. You must resolve these conflicts or accept the proposed resolutions in order to complete this operation. See the topic [Resolving Conflicts](#) for more information.

Note: If you always update your libraries when you launch an interactive workbench session or publish when you close one, you are less likely to have libraries that are out of sync. See the topic [Sharing Libraries](#) for more information.

To add a library

1. From the menus, choose Resources > Add Library. The Add Library dialog box opens.
 2. Select the library or libraries in the list.
 3. Click Add. If any conflicts occur between the newly added libraries and any libraries that were already there, you will be asked to verify the conflict resolutions or change them before completing the operation. See the topic [Resolving Conflicts](#) for more information.
-

Finding Terms and Types

You can search in the various panes in the editor using the Find feature. In the editor, you can choose Edit > Find from the menus and the Find toolbar appears. You can use this toolbar to find one occurrence at a time. By clicking Find again, you can find subsequent occurrences of your search term.

When searching, the editor searches only the library or libraries listed in the drop-down list on the Find toolbar. If All Libraries is selected, the program will search everything in the editor.

When you start a search, it begins in the area that has the focus. The search continues through each section, looping back around until it returns to the active cell. You can reverse the order of the search using the directional arrows. You can also choose whether or not your search is case sensitive.

To Find Strings in the View

1. From the menus, choose Edit > Find. The Find toolbar is displayed.
2. Enter the string for which you want to search.
3. Click the Find button to begin the search. The next occurrence of the term or type is then highlighted.
4. Click the button again to move from occurrence to occurrence.

Using an asterisk in terms

Using an asterisk (*) in terms is especially useful if you are dealing with an agglutinative language that creates new words by compounding other words together without intervening spaces. For example, the German word *Übernachtungspreis*, which is made up of: *Übernachtung* + *s* + *Preis*.

As an example, if you search in terms for *preis** in the type **Budget**, it will match extracted concepts such as *preiserhöhung*. In the same way, **preis* will match *Übernachtung* and **preis** will match *Übernachtungspreiserhöhung*.

Related information

- [Backing Up Resources](#)
- [Importing resource files](#)
- [Working with Libraries](#)

- [Shipped libraries](#)
 - [Creating Libraries](#)
 - [Adding public libraries](#)
 - [Viewing Libraries](#)
 - [Managing Local Libraries](#)
 - [Managing Public Libraries](#)
 - [Sharing Libraries](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Viewing Libraries

You can display the contents of one particular library or all libraries. This can be helpful when dealing with many libraries or when you want to review the contents of a specific library before publishing it. Changing the view only impacts what you see in this Library Resources tab but does not disable any libraries from being used during extraction. See the topic [Disabling Local Libraries](#) for more information.

The default view is All Libraries, which shows all libraries in the tree and their contents in other panes. You can change this selection using the drop-down list on the toolbar or through a menu selection (View > Libraries) When a single library is being viewed, all items in other libraries disappear from view but are still read during the extraction.

To Change the Library View

1. From the menus in the Library Resources tab, choose View > Libraries. A menu with all of the local libraries opens.
2. Select the library that you want to see or select the All Libraries option to see the contents of all libraries. The contents of the view are filtered according to your selection.

Related information

- [Backing Up Resources](#)
 - [Importing resource files](#)
 - [Working with Libraries](#)
 - [Shipped libraries](#)
 - [Creating Libraries](#)
 - [Adding public libraries](#)
 - [Finding Terms and Types](#)
 - [Managing Local Libraries](#)
 - [Managing Public Libraries](#)
 - [Sharing Libraries](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Managing Local Libraries

Local libraries are the libraries inside your interactive workbench session or inside a template, as opposed to public libraries. See the topic [Managing Public Libraries](#) for more information. There are also some basic local library management tasks that you might want to perform, including:

- Renaming a local library. See the topic [Renaming Local Libraries](#) for more information.
 - Disabling or enabling a local library. See the topic [Disabling Local Libraries](#) for more information.
 - Deleting a local library. See the topic [Deleting Local Libraries](#) for more information.
- [Renaming Local Libraries](#)
 - [Disabling Local Libraries](#)
 - [Deleting Local Libraries](#)

Related information

- [Backing Up Resources](#)
- [Importing resource files](#)
- [Working with Libraries](#)
- [Shipped libraries](#)
- [Creating Libraries](#)
- [Adding public libraries](#)
- [Finding Terms and Types](#)
- [Viewing Libraries](#)

- [Managing Public Libraries](#)
 - [Sharing Libraries](#)
 - [Microsoft Internet Explorer settings for Help](#)
 - [Renaming Local Libraries](#)
 - [Disabling Local Libraries](#)
 - [Deleting Local Libraries](#)
-

Renaming Local Libraries

You can rename local libraries. If you rename a local library, you will disassociate it from the public version, if a public version exists. This means that subsequent changes can no longer be shared with the public version. You can republish this local library under its new name. This also means that you will not be able to update the original public version with any changes that you make to this local version.

Note: You cannot rename a public library.

1. From the menus, choose Edit > Library Properties. The Library Properties dialog box opens.

To Rename a Local Library

1. In the tree view, select the library that you want to rename.
2. Enter a new name for the library in the Name text box.
3. Click OK to accept the new name for the library. The dialog box closes and the library name is updated in the tree view.

Related information

- [Managing Local Libraries](#)
 - [Disabling Local Libraries](#)
 - [Deleting Local Libraries](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Disabling Local Libraries

If you want to temporarily exclude a library from the extraction process, you can deselect the check box to the left of the library name in the tree view. This signals that you want to keep the library but want the contents ignored when checking for conflicts and during extraction.

To Disable a Library

1. In the library tree pane, select the library you want to disable.
2. Click the spacebar. The check box to the left of the name is cleared.

Related information

- [Managing Local Libraries](#)
 - [Renaming Local Libraries](#)
 - [Deleting Local Libraries](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Deleting Local Libraries

You can remove a library without deleting the public version of the library and vice versa. Deleting a local library will delete the library and all of its content from session only. Deleting a local version of a library does not remove that library from other sessions or the public version. See the topic [Managing Public Libraries](#) for more information.

To Delete a Local Library

1. In the tree view, select the library you want to delete.
2. From the menus, choose Edit > Delete to delete the library. The library is removed.
3. If you have never published this library before, a message asking whether you would like to delete or keep this library opens. Click Delete to continue or Keep if you would like to keep this library.

Note: One library must always remain.

Related information

- [Managing Local Libraries](#)
 - [Renaming Local Libraries](#)
 - [Disabling Local Libraries](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Managing Public Libraries

In order to reuse local libraries, you can publish them and then work with them and see them through the Manage Libraries dialog box (Resources > Manage Libraries). See the topic [Sharing Libraries](#) for more information. Some basic public library management tasks that you might want to perform include importing, exporting, or deleting a public library. You cannot rename a public library.

Importing Public Libraries

1. In the Manage Libraries dialog box, click Import.... The Import Library dialog box opens.
2. Select the library file (*.lib) that you want to import and if you also want to add this library locally, select Add library to current project.
3. Click Import. The dialog box closes. If a public library with the same name already exists, you will be asked to rename the library that you are importing or to overwrite the current public library.

Exporting Public Libraries

You can export public libraries into the .lib format so that you can share them.

1. In the Manage Libraries dialog box, select the library that you want to export in the list.
2. Click Export. The Select Directory dialog box opens.
3. Select the directory to which you want to export and click Export. The dialog box closes and the library file (*.lib) is exported.

Deleting Public Libraries

You can remove a local library without deleting the public version of the library and vice versa. However, if the library is deleted from this dialog box, it can no longer be added to any session resources until a local version is published again.

If you delete a library that was installed with the product, the originally installed version is restored.

1. In the Manage Libraries dialog box, select the library that you want to delete. You can sort the list by clicking on the appropriate header.
2. Click Delete to delete the library. IBM® SPSS® Modeler Text Analytics verifies whether the local version of the library is the same as the public library. If so, the library is removed with no alert. If the library versions differ, an alert opens to ask you whether you want to keep or remove the public version is issued.

Related information

- [Backing Up Resources](#)
 - [Importing resource files](#)
 - [Working with Libraries](#)
 - [Shipped libraries](#)
 - [Creating Libraries](#)
 - [Adding public libraries](#)
 - [Finding Terms and Types](#)
 - [Viewing Libraries](#)
 - [Managing Local Libraries](#)
 - [Sharing Libraries](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Sharing Libraries

Libraries allow you to work with resources in a way that is easy to share among multiple interactive workbench sessions. Libraries can exist in two states, or versions. Libraries that are editable in the editor and part of an interactive workbench session are called **local libraries**. While working with in an interactive workbench session, you may make a lot of changes in the *Vegetables* library, for example. If your changes could be useful with other data, you can make these resources available by creating a **public library** version of the *Vegetables* library. A public library, as the name implies, is available to any other resources in any interactive workbench session.

You can see the public libraries in the Manage Libraries dialog box. Once this public library version exists, you can add it to the resources in other contexts so that these custom linguistic resources can be shared.

The shipped libraries are initially public libraries. It is possible to edit the resources in these libraries and then create a new public version. Those new versions would then be accessible in other interactive workbench sessions.

As you continue to work with your libraries and make changes, your library versions will become desynchronized. In some cases, a local version might be more recent than the public version, and in other cases, the public version might be more recent than the local version. It is also possible for both the public and local versions to contain changes that the other does not if the public version was updated from within another interactive workbench session. If your library versions become desynchronized, you can synchronize them again. Synchronizing library versions consists of republishing and/or updating local libraries.

Whenever you launch an interactive workbench session or close one, you will be prompted to synchronize any libraries that need updating or republishing. Additionally, you can easily identify the synchronization state of your local library by the icon appearing beside the library name in the tree view or by viewing the Library Properties dialog box. You can also choose to do so at any time through menu selections. The following table describes the five possible states and their associated icons.

Table 1. Local library synchronization states

Icon	Local library status description
	Unpublished—The local library has never been published.
	Synchronized—The local and public library versions are identical. This also applies to the <i>Local Library</i> , which cannot be published because it is intended to contain only session-specific resources.
	Out of date—The public library version is more recent than the local version. You can update your local version with the changes.
	Newer—The local library version is more recent than the public version. You can republish your local version to the public version.
	Out of sync—Both the local and public libraries contain changes that the other does not. You must decide whether to update or publish your local library. If you update, you will lose the changes that you made since the last time you updated or published. If you choose to publish, you will overwrite the changes in the public version.

Note: If you always update your libraries when you launch an interactive workbench session or publish when you close one, you are less likely to have libraries that are out of synchronization.

You can republish a library any time you think that the changes in the library would benefit other streams that may also contain this library. Then, if your changes would benefit other streams, you can update the local versions in those streams. In this way, you can create streams for each context or domain that applies to your data by creating new libraries and/or adding any number of public libraries to your resources.

If a public version of a library is shared, there is a greater chance that differences between local and public versions will arise. Whenever you launch or close and publish from an interactive workbench session or open or close a template from the Template Editor, a message is displayed to enable you to publish and/or update any libraries whose versions are not in sync with those in the Manage Libraries dialog box. If the public library version is more recent than the local version, a dialog box asking whether you would like to update opens. You can choose whether to keep the local version as is instead of updating with the public version or merge the updates into the local library.

- [Publishing Libraries](#)
- [Updating Libraries](#)

Related information

- [Publishing Libraries](#)
- [Updating Libraries](#)
- [Resolving Conflicts](#)
- [Backing Up Resources](#)
- [Importing resource files](#)
- [Working with Libraries](#)
- [Shipped libraries](#)
- [Creating Libraries](#)
- [Adding public libraries](#)
- [Finding Terms and Types](#)
- [Viewing Libraries](#)
- [Managing Local Libraries](#)
- [Managing Public Libraries](#)
- [Microsoft Internet Explorer settings for Help](#)

Publishing Libraries

If you have never published a particular library, publishing entails creating a public copy of your local library in the database. If you are republishing a library, the contents of the local library will replace the existing public version's contents. After republishing, you can update this library in any other stream sessions so that their local versions are in sync with the public version. Even though you can publish a library, a local version is always stored in the session.

Important! If you make changes to your local library and, in the meantime, the public version of the library was also changed, your library is considered to be out of sync. We recommend that you begin by updating the local version with the public changes, make any changes that you want, and then publish your local version again to make both versions identical. If you make changes and publish first, you will overwrite any changes in the public version.

To Publish Local Libraries to the Database

1. From the menus, choose Resources > Publish Libraries. The Publish Libraries dialog box opens, with all libraries in need of publishing selected by default.
2. Select the check box to the left of each library that you want to publish or republish.
3. Click Publish to publish the libraries to the Manage Libraries database.

Related information

- [Sharing Libraries](#)
- [Updating Libraries](#)
- [Resolving Conflicts](#)
- [Microsoft Internet Explorer settings for Help](#)

Updating Libraries

Whenever you launch or close an interactive workbench session , you can update or publish any libraries that are no longer in sync with the public versions. If the public library version is more recent than the local version, a dialog box asking whether you would like to update the library opens. You can choose whether to keep the local version instead of updating with the public version or replacing the local version with the public one. If a public version of a library is more recent than your local version, you can update the local version to synchronize its content with that of the public version. Updating means incorporating the changes found in the public version into your local version.

Note: If you always update your libraries when you launch an interactive workbench session or publish when you close one , you are less likely to have libraries that are out of sync. See the topic [Sharing Libraries](#) for more information.

To Update Local Libraries

1. From the menus, choose Resources > Update Libraries. The Update Libraries dialog box opens, with all libraries in need of updating selected by default.
2. Select the check box to the left of each library that you want to publish or republish.
3. Click Update to update the local libraries.

Related information

- [Sharing Libraries](#)
- [Publishing Libraries](#)
- [Resolving Conflicts](#)
- [Microsoft Internet Explorer settings for Help](#)

Resolving Conflicts

Local versus Public Library Conflicts

Whenever you launch a stream session , IBM® SPSS® Modeler Text Analytics performs a comparison of the local libraries and those listed in the Manage Libraries dialog box. If any local libraries in your session are not in sync with the published versions, the Library Synchronization Warning dialog box opens. You can choose from the following options to select the library versions that you want to use here:

- **All libraries local to file.** This option keeps all of your local libraries as they are. You can always republish or update them later.
- **All published libraries on this machine.** This option will replace the shown local libraries with the versions found in the database.
- **All more recent libraries.** This option will replace any older local libraries with the more recent public versions from the database.
- **Other.** This option allows you to manually select the versions that you want by choosing them in the table.

Forced Term Conflicts

Whenever you add a public library or update a local library, conflicts and duplicate entries may be uncovered between the terms and types in this library and the terms and types in the other libraries in your resources. If this occurs, you will be asked to verify the proposed conflict resolutions or change them before completing the operation in the Edit Forced Terms dialog box. See the topic [Forcing terms](#) for more information.

The Edit Forced Terms dialog box contains each pair of conflicting terms or types. Alternating background colors are used to visually distinguish each conflict pair. These colors can be changed in the Options dialog box. See the topic [Options: Display Tab](#) for more information. The Edit Forced Terms dialog box contains two tabs:

- **Duplicates.** This tab contains the duplicated terms found in the libraries. If a pushpin icon appears after a term, it means that this occurrence of the term has been forced. If a black X icon appears, it means that this occurrence of the term will be ignored during extraction because it has been forced elsewhere.
- **User Defined.** This tab contains a list of any terms that have been forced manually in the type dictionary term pane and not through conflicts.

Note: The Edit Forced Terms dialog box opens after you add or update a library. If you cancel out of this dialog box, you will not be canceling the update or addition of the library.

To Resolve Conflicts

1. In the Edit Forced Terms dialog box, select the radio button in the Use column for the term that you want to force.
2. When you have finished, click OK to apply the forced terms and close the dialog box. If you click Cancel, you will cancel the changes you made in this dialog box.

Related information

- [Sharing Libraries](#)
 - [Publishing Libraries](#)
 - [Updating Libraries](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

About Library Dictionaries

The resources used to extract text data are stored in the form of templates and libraries. A library can be made up of three dictionaries.

- The **type dictionary** contains a collection of terms grouped under one label, or type name. When the extraction engine reads your text data, it compares the words found in the text to the terms defined in your type dictionaries. During extraction, inflected forms of a type's terms and synonyms are grouped under a target term called concept. Extracted concepts are assigned to the type dictionary in which they appear as terms. You can manage your type dictionaries in the upper left and center panes of the editor—the library tree and the term pane. See the topic [Type dictionaries](#) for more information.
- The **substitution dictionary** contains a collection of words defined as synonyms or as optional elements used to group similar terms under one target term, called a concept in the final extraction results. You can manage your substitution dictionaries in the lower left pane of the editor using the Synonyms tab and the Optional tab. See the topic [Substitution/Synonym dictionaries](#) for more information.
- The **exclude dictionary** contains a collection of terms and types that will be removed from the final extraction results. You can manage your exclude dictionaries in the rightmost pane of the editor. See the topic [Exclude dictionaries](#) for more information.

See the topic [Working with Libraries](#) for more information.

- [Type dictionaries](#)
- [Substitution/Synonym dictionaries](#)
- [Exclude dictionaries](#)

Related information

- [Type dictionaries](#)
 - [Substitution/Synonym dictionaries](#)
 - [Exclude dictionaries](#)
 - [Microsoft Internet Explorer settings for Help](#)
 - [Session Resource Editor](#)
 - [Templates and Resources](#)
 - [Working with Libraries](#)
 - [About Advanced Resources](#)
 - [About Text Link Rules](#)
-

Type dictionaries

A *type dictionary* is made up of a type name, or label, and a list of terms. Type dictionaries are managed in the upper left and center panes of Library Resources tab in the editor. You can access this view with View > Resource Editor in the menus, if you are in an interactive workbench

session. Otherwise, you can edit dictionaries for a specific template in the Template Editor.

When the extraction engine reads your text data, it compares words found in the text to the terms defined in your type dictionaries. Terms are words or phrases in the type dictionaries in your linguistic resources.

When a word matches a term, it is assigned to the type name for that term. When the resources are read during extraction, the terms that were found in the text then go through several processing steps before they become concepts in the Extraction Results pane. If multiple terms belonging to the same type dictionary are determined to be synonymous by the extraction engine, then they are grouped under the most frequently occurring term and called a *concept* in the Extraction Results pane. For example, if the terms **question** and **query** might appear under the concept name **question** in the end.

Figure 1. Library tree and term pane

The screenshot shows the SPSS Modeler Text Analytics interface. On the left, the 'Library Resources' pane displays a tree view of type dictionaries. Under 'Local Library', there are several checked entries: 'Product Satisfaction Opinions (English)', 'Product Satisfaction Library (English)', 'Opinions Library (English)', 'Budget Library (English)', 'Core Library (English)', and 'Variations Library (English)'. The 'Product Satisfaction Library (English)' node has its own expanded tree with terms like 'after taste', 'age', 'appearance', etc. The 'Term' column lists the words from the type dictionary, 'Match' indicates the matching option (e.g., 'Entire Term', 'Entire And Any'), 'Inflect' shows if inflected forms are generated, 'Type' categorizes the term (e.g., 'Characteristics'), and 'Library' shows the source type dictionary for each term.

Term	Match	Inflect	Type	Library
after taste	Entire Term	✓	Characteristics	Product Satisfaction Library (English)
after-taste	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
aftertaste	Entire Term	✓	Characteristics	Product Satisfaction Library (English)
age	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
appearance	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
aroma	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
attribute	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
audio	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
behaviour	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
can carry	Entire Term	✗	Characteristics	Product Satisfaction Library (English)
can store	Entire Term	✗	Characteristics	Product Satisfaction Library (English)
capacity	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
characteristic	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
characteristic	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
color	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
coloring	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
colour	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
colouring	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
comfort	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
component	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
comfort	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)
consistence	Entire And Any	✓	Characteristics	Product Satisfaction Library (English)

The list of type dictionaries is shown in the library tree pane on the left. The content of each type dictionary appears in the center pane. Type dictionaries consist of more than just a list of terms. The manner in which words and word phrases in your text data are matched to the terms defined in the type dictionaries is determined by the match option defined. A **match option** specifies how a term is anchored with respect to a candidate word or phrase in the text data. See the topic [Adding terms](#) for more information.

Additionally, you can extend the terms in your type dictionary by specifying whether you want to automatically generate and add inflected forms of the terms to the dictionary. By generating the inflected forms, you automatically add plural forms of singular terms, singular forms of plural terms, and adjectives to the type dictionary. See the topic [Adding terms](#) for more information.

Note: For most languages, concepts that are not found in any type dictionary but are extracted from the text are automatically typed as **<Unknown>**

Using an asterisk in terms

Using an asterisk (*) in terms is especially useful if you are dealing with an agglutinative language that creates new words by compounding other words together without intervening spaces. For example, the German word *Übernachtungspreis*, which is made up of: *Übernachtung* + s + *Preis*.

As an example, if you search in terms for **preis*** in the type **Budget**, it will match extracted concepts such as **preiserhöhung**. In the same way, ***preis** will match **Übernachtung** and ***preis*** will match **Übernachtungspreiserhöhung**.

- [Built-in types](#)
- [Creating types](#)
- [Adding terms](#)
- [Forcing terms](#)
- [Renaming types](#)
- [Moving types](#)
- [Disabling and deleting types](#)

Built-in types

IBM® SPSS® Modeler Text Analytics is delivered with a set of linguistic resources in the form of shipped libraries and compiled resources. The shipped libraries contain a set of built-in type dictionaries such as **<Location>**, **<Organization>**, **<Person>**, and **<Product>**.

These type dictionaries are used by the extraction engine to assign types to the concepts it extracts such as assigned the type **<Location>** to the concept **paris**. Although a large number of terms have been defined in the built-in type dictionaries, they do not cover every possibility. Therefore, you can add to them or create your own. For a description of the contents of a particular shipped type dictionary, read the annotation in the Type Properties dialog box. Select the type in the tree and choose Edit > Properties from the context menu.

Note:

In addition to the shipped libraries, the compiled resources (also used by the extraction engine) contain a large number of definitions complementary to the built-in type dictionaries, but their content is not visible in the product. You can, however, force a term that was typed by the compiled dictionaries into any other dictionary. See [Forcing terms](#) for more information.

Creating types

You can create type dictionaries to help group similar terms. When terms appearing in this dictionary are discovered during the extraction process, they will be assigned to this type name and extracted under a concept name. Whenever you create a library, an empty type library is always included so that you can begin entering terms immediately.

If you are analyzing text about food and want to group terms relating to vegetables, you could create your own <Vegetables> type dictionary. You could then add terms such as **carrot**, **broccoli**, and **spinach** if you feel that they are important terms that will appear in the text. Then, during extraction, if any of these terms are found, they are extracted as concepts and assigned to the <Vegetables> type.

You do not have to define every form of a word or expression, because you can choose to generate the inflected forms of terms. By choosing this option, the extraction engine will automatically recognize singular or plural forms of the terms among other forms as belonging to this type. This option is particularly useful when your type contains mostly nouns, since it is unlikely you would want inflected forms of verbs or adjectives.

The Type Properties dialog box contains the following fields.

Name. The name you give to the type dictionary you are creating. We recommend that you do not use spaces in type names, especially if two or more type names start with the same word.

Note: There are some constraints about type names and the use of symbols. For example, do not use symbols such as "@" or "!" within the name. Default match. The default match attribute instructs the extraction engine how to match this term to text data. Whenever you add a term to this type dictionary, this is the match attribute automatically assigned to it. You can always change the match choice manually in the term list. Options include: Entire Term, Start, End, Any, Start or End, Entire and Start, Entire and End, Entire and (Start or End), and Entire (no compounds). See the topic [Adding terms](#) for more information.

Add to. This field indicates the library in which you will create your new type dictionary.

Generate inflected forms by default. This option tells the extraction engine to use grammatical morphology to capture and group similar forms of the terms that you add to this dictionary, such as singular or plural forms of the term. This option is particularly useful when your type contains mostly nouns. When you select this option, all new terms added to this type will automatically have this option although you can change it manually in the list.

Font color. This field allows you to distinguish the results from this type from others in the interface. If you select Use parent color, the default type color is used for this type dictionary, as well. This default color is set in the options dialog box. See the topic [Options: Display Tab](#) for more information. If you select Custom, select a color from the drop-down list.

Annotation. This field is optional and can be used for any comments or descriptions.

To create a type dictionary

1. Select the library in which you would like to create a new type dictionary.
2. From the menus, choose Tools > New Type. The Type Properties dialog box opens.
3. Enter the name of your type dictionary in the Name text box and choose the options you want.
4. Click OK to create the type dictionary. The new type is visible in the library tree pane and appears in the center pane. You can begin adding terms immediately. For more information, see [Adding terms](#).

Note: These instructions show you how to make changes within the Resource Editor view or the Template Editor . Keep in mind that you can also do this kind of fine-tuning directly from the Extraction Results pane , Data pane, Categories pane, or Cluster Definitions dialog box in the other views. See the topic [Refining extraction results](#) for more information.

Adding terms

The library tree pane displays libraries and can be expanded to show the type dictionaries that they contain. In the center pane, a term list displays the terms in the selected library or type dictionary, depending on the selection in the tree.

In the Resource Editor, you can add terms to a type dictionary directly in the term pane or through the Add New Terms dialog box. The terms that you add can be single words or compound words. You will always find a blank row at the top of the list to allow you to add a new term.

Note: These instructions show you how to make changes within the Resource Editor view or the Template Editor . Keep in mind that you can also do this kind of fine-tuning directly from the Extraction Results pane , Data pane, Categories pane, or Cluster Definitions dialog box in the other views. See the topic [Refining extraction results](#) for more information.

Term column

In this column, enter single or compound words into the cell. The color in which the term appears depends on the color for the type in which the term is stored or forced. You can change type colors in the Type Properties dialog box. See the topic [Creating types](#) for more information.

Force column

In this column, by putting a pushpin icon into this cell, the extraction engine knows to ignore any other occurrences of this same term in other libraries. See the topic [Forcing terms](#) for more information.

Match column

In this column, select a match option to instruct the extraction engine how to match this term to text data. See the table for examples. You can change the default value by editing the type properties. See the topic [Creating types](#) for more information. From the menus, choose Edit > Change Match. The following are the basic match options since combinations of these are also possible:

- Start. If the term in the dictionary matches the first word in a concept extracted from the text, this type is assigned. For example, if you enter **apple**, **apple tart** will be matched.
- End. If the term in the dictionary matches the last word in a concept extracted from the text, this type is assigned. For example, if you enter **apple**, **cider apple** will be matched.
- Any. If the term in the dictionary matches any word of a concept extracted from the text, this type is assigned. For example, if you enter **apple**, the Any option will type **apple tart**, **cider apple**, and **cider apple tart** the same way.
- Entire Term. If the entire concept extracted from the text matches the exact term in the dictionary, this type is assigned. Adding a term as Entire term, Entire and Start, Entire and End, Entire and Any, or Entire (no compounds) will force the extraction of a term. Furthermore, since the **<Person>** type extracts only two part names, such as *edith piaf* or *mohandas gandhi*, you may want to explicitly add the first names to this type dictionary if you are trying to extract a first name when no last name is mentioned. For example, if you want to catch all instances of *edith* as a name, you should add **edith** to the **<Person>** type using Entire term or Entire and Start.
- Entire (no compounds). If the entire concept extracted from the text matches the exact term in the dictionary, this type is assigned and the extraction is stopped to prohibit the extraction from matching the term to a longer compound. For example, if you enter **apple**, the Entire (no compound) option will type **apple** and not extract the compound **apple sauce** unless it is forced in somewhere else.

In the following table, assume that the term **apple** is in a type dictionary. Depending on the match option, this table shows which concepts would be extracted and typed if they were found in the text.

Table 1. Matching Examples

Match options for the term: 	Extracted concepts			
	apple	apple tart	ripe apple	homemade apple tart
Entire Term	✓			
Start		✓		
End			✓	
Start or End		✓	✓	
Entire and Start	✓	✓		
Entire and End	✓		✓	
Entire and (Start or End)	✓	✓	✓	
Any		✓	✓	✓
Entire and Any	✓	✓	✓	✓
Entire (no compounds)	✓	never extracted	never extracted	never extracted

Inflect column

In this column, select whether the extraction engine should generate inflected forms of this term during extraction so that they are all grouped together. The default value for this column is defined in the Type Properties but you can change this option on a case-by-case basis directly in the column. From the menus, choose Edit > Change Inflection.

Type column

In this column, select a type dictionary from the drop-down list. The list of types is filtered according to your selection in the library tree pane. The first type in the list is always the default type selected in the library tree pane. From the menus, choose Edit > Change Type.

Library column

In this column, the library in which your term is stored appears. You can drag and drop a term into another type in the library tree pane to change its library.

To add a single term to a type dictionary

1. In the library tree pane, select the type dictionary to which you want to add the term.
2. In the term list in the center pane, type your term in the first available empty cell and set any options you want for this term.

To add multiple terms to a type dictionary

1. In the library tree pane, select the type dictionary to which you want to add terms.
2. From the menus, choose Tools > New Terms. The Add New Terms dialog box opens.
3. Enter the terms you want to add to the selected type dictionary by typing the terms or copying and pasting a set of terms. If you enter multiple terms, you must separate them using the delimiter that is defined in the Options dialog, or add each term on a new line. See the topic [Setting Options](#) for more information.
4. Click OK to add the terms to the dictionary. The match option is automatically set to the default option for this type library. The dialog box closes and the new terms appear in the dictionary.

Forcing terms

If you want a term to be assigned to a particular type, you can add it to the corresponding type dictionary. However, if there are multiple terms with the same name, the extraction engine must know which type should be used. Therefore, you will be prompted to select which type should be used. This is called *forcing* a term into a type. This option is most useful when overriding the type assignment from a compiled (internal, non-editable) dictionary. In general, we recommend avoiding duplicate terms altogether.

Forcing will not *remove* the other occurrences of this term; rather, they will be ignored by the extraction engine. You can later change which occurrence should be used by forcing or unforcing a term. You may also need to force a term into a type dictionary when you add a public library or update a public library.

You can see which terms are forced or ignored in the Force column, the second column in the term pane. If a pushpin icon appears, this means that this occurrence of the term has been forced. If a black X icon appears, this means that this occurrence of the term will be ignored during extraction because it has been forced elsewhere. Additionally, when you force a term, it will appear in the color for the type in which it was forced. This means that if you forced a term that is in both **Type 1** and **Type 2** into **Type 1**, any time you see this term in the window, it will appear in the font color defined for **Type 1**.

1.

You can double-click the icon in order to change the status. If the term appears elsewhere, a Resolve Conflicts dialog box opens to allow you to select which occurrence should be used.

Renaming types

You can rename a type dictionary or change other dictionary settings by editing the type properties.

Important: We recommend that you do not use spaces in type names, especially if two or more type names start with the same word. We also recommend that you do not rename the types in the Core or Opinions libraries or change their default match attributes.

To rename a type

1. In the library tree pane, select the type dictionary you want to rename.
2. Right-click your mouse and choose Type Properties from the context menu. The Type Properties dialog box opens.
3. Enter the new name for your type dictionary in the Name text box.
4. Click OK to accept the new name. The new type name is visible in the library tree pane.

Moving types

You can drag a type dictionary to another location within a library or to another library in the tree.

To reorder a type within a library

1. In the library tree pane, select the type dictionary you want to move.
2. From the menus, choose Edit > Move Up to move the type dictionary up one position in the library tree pane or Edit > Move Down to move it down one position.

To move a type to another library

1. In the library tree pane, select the type dictionary you want to move.
2. Right-click your mouse and choose Type Properties from the context menu. The Type Properties dialog box opens. (You can also drag and drop the type into another library).
3. In the Add To list box, select the library to which you want to move the type dictionary.
4. Click OK. The dialog box closes, and the type is now in the library you selected.

Disabling and deleting types

If you want to temporarily remove a type dictionary, you can disable it by deselecting the check box to the left of the dictionary name in the library tree pane. This signals that you want to keep the dictionary in your library but want the contents ignored during conflict checking and during the extraction process.

You can also permanently delete type dictionaries from a library.

To disable a type dictionary

1. In the library tree pane, select the type dictionary you want to disable.
2. Click the spacebar. The check box to the left of the type name is cleared.

To delete a type dictionary

1. In the library tree pane, select the type dictionary you want to delete.
2. From the menus, choose Edit > Delete to delete the type dictionary.

Substitution/Synonym dictionaries

A *substitution dictionary* is a collection of terms that help to group similar terms under one target term. Substitution dictionaries are managed in the bottom pane of the Library Resources tab. You can access this view with View > Resource Editor in the menus, if you are in an interactive workbench session. Otherwise, you can edit dictionaries for a specific template in the Template Editor.

You can define two forms of substitutions in this dictionary: *synonyms* and *optional elements*. You can click the tabs in this pane to switch between them.

After you run an extraction on your text data, you may find several concepts that are synonyms or inflected forms of other concepts. By identifying optional elements and synonyms, you can force the extraction engine to map these to one single target term.

Substituting using synonyms and optional elements reduces the number of concepts in the Extraction Results pane by combining them together into more significant, representative concepts with higher frequency Doc. counts.

Synonyms

Synonyms associate two or more words that have the same meaning. You can also use synonyms to group terms with their abbreviations or to group commonly misspelled words with the correct spelling. You can define these synonyms on the Synonyms tab.

A synonym definition is made up of two parts. The first is a Target term, which is the term under which you want the extraction engine to group all synonym terms. Unless this target term is used as a synonym of another target term or unless it is excluded, it is likely to become the concept that appears in the Extraction Results pane. The second is the list of synonyms that will be grouped under the target term.

For example, if you want **automobile** to be replaced by **vehicle**, then **automobile** is the synonym and **vehicle** is the target term.

You can enter any words into the Synonym column, but if the word is not found during extraction and the term had a match option with **Entire**, then no substitution can take place. However, the target term does not need to be extracted for the synonyms to be grouped under this term.

Optional elements

Optional elements identify optional words in a compound term that can be ignored during extraction in order to keep similar terms together even if they appear slightly different in the text. Optional elements are single words that, if removed from a compound, could create a match with another term. These single words can appear anywhere within the compound--at the beginning, middle, or end. You can define optional elements on the Optional tab.

For example, to group the terms **ibm** and **ibm corp** together, you should declare **corp** to be treated as an optional element in this case. In another example, if you designate the term **access** to be an optional element and during extraction both **internet access speed** and **internet speed** are found, they will be grouped together under the term that occurs most frequently.

- [Defining synonyms](#)
 - [Defining optional elements](#)
 - [Disabling and Deleting Substitutions](#)
-

Defining synonyms

On the Synonyms tab, you can enter a synonym definition in the empty line at the top of the table. Begin by defining the target term and its synonyms. You can also select the library in which you would like to store this definition. During extraction, all occurrences of the synonyms will be grouped under the target term in the final extraction. See the topic [Adding terms](#) for more information.

For example, if your text data includes a lot of telecommunications information, you may have these terms: **cellular phone**, **wireless phone**, and **mobile phone**. In this example, you may want to define **cellular** and **mobile** as synonyms of **wireless**. If you define these synonyms, then every extracted occurrence of **cellular phone** and **mobile phone** will be treated as the same term as **wireless phone** and will appear together in the term list.

When you are building your type dictionaries, you may enter a term and then think of three or four synonyms for that term. In that case, you could enter all of the terms and then your target term into the substitution dictionary and then drag the synonyms.

Synonym substitution is also applied to the inflected forms (such as the plural form) of the synonym. Depending on the context, you may want to impose constraints on how terms are substituted. Certain characters can be used to place limits on how far the synonym processing should go:

- Exclamation mark (!). When the exclamation mark directly precedes the synonym **!synonym**, this indicates that no inflected forms of the synonym will be substituted by the target term. However, an exclamation mark directly preceding the target term **!target-term** means that you do not want any part of the compound target term or variants to receive any further substitutions.
- Asterisk (*). An asterisk placed directly after a synonym, such as **synonym***, means that you want this word to be replaced by the target term. For example, if you defined **manage*** as the synonym and **management** as the target, then **associate managers** will be replaced by the target term **associate management**. You can also add a space and an asterisk after the word **(synonym *)** such as **internet ***. If you defined the target as **internet** and the synonyms as **internet * *** and **web ***, then **internet access card** and **web portal** would be replaced with **internet**. You cannot begin a word or string with the asterisk wildcard in this dictionary.
- Caret (^). A caret and a space preceding the synonym, such as **^ synonym**, means that the synonym grouping applies only when the term begins with the synonym. For example, if you define **^ wage** as the synonym and **income** as the target and both terms are extracted, then they will be grouped together under the term **income**. However, if **minimum wage** and **income** are extracted, they will not be grouped together, since **minimum wage** does not begin with **wage**. A space must be placed between this symbol and the synonym.
- Dollar sign (\$). A space and a dollar sign following the synonym, such as **synonym \$**, means that the synonym grouping applies only when the term ends with the synonym. For example, if you define **cash \$** as the synonym and **money** as the target and both terms are extracted, then they will be grouped together under the term **money**. However, if **cash cow** and **money** are extracted, they will not be grouped together, since **cash cow** does not end with **cash**. A space must be placed between this symbol and the synonym.
- Caret (^) and dollar sign (\$). If the caret and dollar sign are used together, such as **^ synonym \$**, a term matches the synonym only if it is an exact match. This means that no words can appear before or after the synonym in the extracted term in order for the synonym grouping to take place. For example, you may want to define **^ van \$** as the synonym and **truck** as the target so that only **van** is grouped with **truck**, while **marie van guerin** will be left unchanged. Additionally, whenever you define a synonym using the caret and dollar signs and this word appears anywhere in the source text, the synonym is automatically extracted.

To add a synonym entry

1. With the substitution pane displayed, click the Synonyms tab in the lower left corner.
2. In the empty line at the top of the table, enter your target term in the Target column. The target term you entered appears in color. This color represents the type in which the term appears or is forced, if that is the case. If the term appears in black, this means that it does not appear in any type dictionaries.
3. Click the second cell to the right of the target and enter the set of synonyms. Separate each entry using the global delimiter as defined in the Options dialog box. See the topic [Setting Options](#) for more information. The terms that you enter appear in color. This color represents the type in which the term appears. If the term appears in black, this means that it does not appear in any type dictionaries.

4. Click the last cell to select the library in which you want to store this synonym definition.

Note: These instructions show you how to make changes within the Resource Editor view or the Template Editor . Keep in mind that you can also do this kind of fine-tuning directly from the Extraction Results pane , Data pane, Categories pane, or Cluster Definitions dialog box in the other views. See the topic [Refining extraction results](#) for more information.

Defining optional elements

On the Optional tab, you can define optional elements for any library you want. These entries are grouped together for each library. As soon as a library is added to the library tree pane, an empty optional element line is added to the Optional tab.

All entries are transformed into lowercase words automatically. The extraction engine will match entries to both lowercase and uppercase words in the text.

Note: Terms are delimited using the delimiter defined in the Options dialog. See the topic [Setting Options](#) for more information. If the optional element that you are entering includes the same delimiter as part of the term, a backslash must precede it.

To add an entry

1. With the substitution pane displayed, click the Optional tab in the lower left corner of the editor.
2. Click in the cell in the Optional Elements column for the library to which you want to add this entry.
3. Enter the optional element. Separate each entry using the global delimiter as defined in the Options dialog box. See the topic [Setting Options](#) for more information.

Disabling and Deleting Substitutions

You can remove an entry in a temporary manner by disabling it in your dictionary. By disabling an entry, the entry will be ignored during extraction.

You can also delete any obsolete entries in your substitution dictionary.

To Disable an Entry

1. In your dictionary, select the entry you want to disable.
2. Click the spacebar. The check box to the left of the entry is cleared.

Note: You can also deselect the check box to the left of the entry to disable it.

To Delete a Synonym Entry

1. In your dictionary, select the entry you want to delete.
2. From the menus, choose Edit > Delete or press the Delete key on your keyboard. The entry is no longer in the dictionary.

To Delete an Optional Element Entry

1. In your dictionary, double-click the entry you want to delete.
2. Manually delete the term.
3. Press Enter to apply the change.

Related information

- [Substitution/Synonym dictionaries](#)
- [Defining synonyms](#)
- [Defining optional elements](#)
- [Microsoft Internet Explorer settings for Help](#)

Exclude dictionaries

An *exclude dictionary* is a list of words, phrases, or partial strings. Any terms matching or containing an entry in the exclude dictionary will be ignored or excluded from extraction. Exclude dictionaries are managed in the right pane of the editor. Typically, the terms that you add to this list are fill-in words or phrases that are used in the text for continuity but that do not really add anything important to the text and may clutter the extraction results. By adding these terms to the exclude dictionary, you can make sure that they are never extracted.

Exclude dictionaries are managed in the upper right pane of Library Resources tab in the editor. You can access this view with View > Resource Editor in the menus, if you are in an interactive workbench session. Otherwise, you can edit dictionaries for a specific template in the Template Editor .

In the exclude dictionary, you can enter a word, phrase, or partial string in the empty line at the top of the table. You can add character strings to your exclude dictionary as one or more words or even partial words using the asterisk as a wildcard. The entries declared in the exclude dictionary will be used to bar concepts from extraction. If an entry is also declared somewhere else in the interface, such as in a type dictionary, it is shown with a strike-through in the other dictionaries, indicating that it is currently excluded. This string does not have to appear in the text data or be declared as part of any type dictionary to be applied.

Note: If you add a concept to the exclude dictionary that also acts as the target in a synonym entry, then the target and all of its synonyms will also be excluded. See the topic [Defining synonyms](#) for more information.

Using wildcards (*)

can use the asterisk wildcard to denote that you want the exclude entry to be treated as a partial string. Any terms found by the extraction engine that contain a word that begins or ends with a string entered in the exclude dictionary will be excluded from the final extraction. However, there are two cases where the wildcard usage is not permitted:

- Dash character (-) preceded by an asterisk wildcard, such as *-
- Apostrophe (') preceded by an asterisk wildcard, such as *'

Table 1. Examples of exclude entries

Entry	Example	Results
word	<i>next</i>	No concepts (or its terms) will be extracted if they contain the word <i>next</i> .
phrase	<i>for example</i>	No concepts (or its terms) will be extracted if they contain the phrase <i>for example</i> .
partial	<i>copyright*</i>	Will exclude any concepts (or its terms) matching or containing the variations of the word <i>copyright</i> , such as copyrighted, copyrighting, copyrights, or copyright 2010 .
partial	<i>*ware</i>	Will exclude any concepts (or its terms) matching or containing the variations of the word <i>ware</i> , such as freeware, shareware, software, hardware, beware, or silverware .

To add entries

- In the empty line at the top of the table, enter a term. The term that you enter appears in color. This color represents the type in which the term appears. If the term appears in black, this means that it does not appear in any type dictionaries.

To disable entries

You can temporarily remove an entry by disabling it in your exclude dictionary. By disabling an entry, the entry will be ignored during extraction.

1. In your exclude dictionary, select the entry that you want to disable.
2. Click the spacebar. The check box to the left of the entry is cleared.

Note: You can also deselect the check box to the left of the entry to disable it.

To delete entries

You can delete any unneeded entries in your exclude dictionary.

1. In your exclude dictionary, select the entry that you want to delete.
2. From the menus, choose Edit > Delete. The entry is no longer in the dictionary.

About Advanced Resources

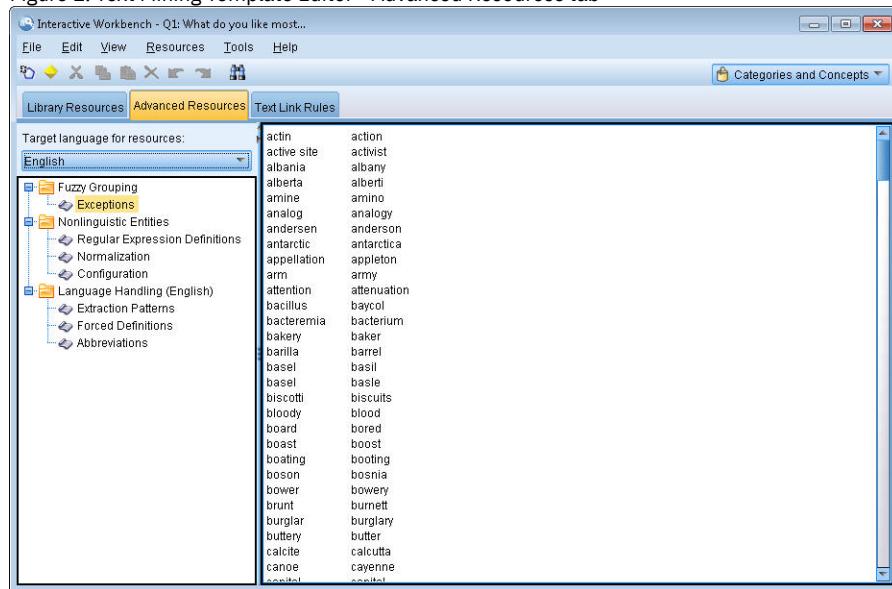
In addition to type, exclude and substitution dictionaries, you can also work with a variety of advanced resource settings such as Fuzzy Grouping settings or nonlinguistic type definitions. You can work with these resources in the Advanced Resources tab in the Template Editor or Resource Editor view.

When you go to the Advanced Resources tab, you can edit the following information:

- Target language for resources. Used to select the language for which the resources will be created and tuned. See the topic [Target Language for Resources](#) for more information.
- Fuzzy Grouping (Exceptions). Used to exclude word pairs from the fuzzy grouping (spelling error correction) algorithm. See the topic [Fuzzy Grouping](#) for more information.

- Nonlinguistic Entities. Used to enable and disable which nonlinguistic entities can be extracted, as well as the regular expressions and the normalization rules that are applied during their extraction. See the topic [Nonlinguistic Entities](#) for more information.
- Language Handling. Used to declare the special ways of structuring sentences (extraction patterns and forced definitions) and using abbreviations for the selected language. See the topic [Language Handling](#) for more information.

Figure 1. Text Mining Template Editor - Advanced Resources tab



Note: You can use the Find/Replace toolbar to find information quickly or to make uniform changes to a section. For more information, see [Replacing](#).

To Edit Advanced Resources

1. Locate and select the resource section that you want to edit. The contents appear in the right pane.
2. Use the menu or the toolbar buttons to cut, copy, or paste content, if necessary.
3. Edit the file(s) that you want to change using the formatting rules in this section. Your changes are saved as soon as you make them. Use the undo or redo arrows on the toolbar to revert to the previous changes.

- [Finding](#)
- [Replacing](#)
- [Target Language for Resources](#)
- [Fuzzy Grouping](#)
- [Nonlinguistic Entities](#)
- [Language Handling](#)

Finding

In some cases, you may need to locate information quickly in a particular section. For example, if you perform text link analysis, you may have hundreds of macros and pattern definitions. Using the Find feature, you can find a specific rule quickly. To search for information in a section, you can use the Find toolbar.

To Use the Find Feature

1. Locate and select the resource section that you want to search. The contents appear in the right pane of the editor.
2. From the menus, choose Edit > Find. The Find toolbar appears at the upper right of the Edit Advanced Resources dialog box.
3. Enter the word string that you want to search for in the text box. You can use the toolbar buttons to control the case sensitivity, partial matching, and direction of the search.
4. Click Find to start the search. If a match is found, the text is highlighted in the window.
5. Click Find again to look for the next match.

Note: When working in the Text Link Rules tab, the Find option is only available when you view the source code.

Related information

- [About Advanced Resources](#)

- [Replacing](#)
 - [Target Language for Resources](#)
 - [Fuzzy Grouping](#)
 - [Nonlinguistic Entities](#)
 - [Language Handling](#)
 - [Viewing and working in source mode](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Replacing

In some cases, you may need to make broader updates to your advanced resources. The Replace feature can help you to make uniform updates to your content.

To Use the Replace Feature

1. Locate and select the resource section in which you want to search and replace. The contents appear in the right pane of the editor.
2. From the menus, choose Edit > Replace. The Replace dialog box opens.
3. In the Find what text box, enter the word string that you want to search for.
4. In the Replace with text box, enter the string that you want to use in place of the text that was found.
5. Select Match whole word only if you want to find or replace only complete words.
6. Select Match case if you want to find or replace only words that match the case exactly.
7. Click Find Next to find a match. If a match is found, the text is highlighted in the window. If you do not want to replace this match, click Find Next again until you find a match that you want to replace.
8. Click Replace to replace the selected match.
9. Click Replace to replace all matches in the section. A message opens with the number of replacements made.
10. When you are finished making your replacements, click Close. The dialog box closes.

Note: If you made a replacement error, you can undo the replacement by closing the dialog box and choosing Edit > Undo from the menus. You must perform this once for every change that you want to undo.

Related information

- [About Advanced Resources](#)
 - [Finding](#)
 - [Target Language for Resources](#)
 - [Fuzzy Grouping](#)
 - [Nonlinguistic Entities](#)
 - [Language Handling](#)
 - [Viewing and working in source mode](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Target Language for Resources

Resources are created for a particular text language. The language for which these resources are tuned is defined in the Advanced Resources tab. You can switch to another language if necessary by selecting that language in the Target language for resources combobox. Additionally, the language listed here will appear as the language for any text analysis packages you create with these resources.

Important: You will rarely ever need to change the language in your resources. Doing so can cause issues when your resources no longer match the extraction language. Though rarely employed, you might change a language if you planned to use the **All** language option during extraction because you expected to have text in more than one language. By changing the language, you can access, for example, the language handling resources for extraction patterns, abbreviations and force definitions for the secondary language you are interested in. However, keep in mind that before publishing or saving the resource changes you've made or running another extraction, set the language back to the primary language you are interested in extracting.

Related information

- [About Advanced Resources](#)
 - [Finding](#)
 - [Replacing](#)
 - [Fuzzy Grouping](#)
 - [Nonlinguistic Entities](#)
-

- [Language Handling](#)
 - [Viewing and working in source mode](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Fuzzy Grouping

In the Text Mining node and Extraction Settings, if you select Accommodate spelling for a minimum root character limit of, you have enabled the fuzzy grouping algorithm.

Fuzzy grouping helps to group commonly misspelled words or closely spelled words by temporarily stripping all vowels (except for the first vowel) and double or triple consonants from extracted words and then comparing them to see if they are the same. During the extraction process, the fuzzy grouping feature is applied to the extracted terms and the results are compared to determine whether any matches are found. If so, the original terms are grouped together in the final extraction list. They are grouped under the term that occurs most frequently in the data.

Note: If the two terms being compared are assigned to different types, excluding the <Unknown> type, then the fuzzy grouping technique is not be applied to this pair. In other words, the terms must belong to the same type or the <Unknown> type in order for the technique to be applied. If you enabled this feature and found that two words with similar spelling were incorrectly grouped together, you may want to exclude them from fuzzy grouping. You can do this by entering the incorrectly matched pairs into the Exceptions section in the Advanced Resources tab. See the topic [About Advanced Resources](#) for more information.

The following example demonstrates how fuzzy grouping is performed. If fuzzy grouping is enabled, these words appear to be the same and are matched in the following manner:

color -> colr	mountain -> montn
colour -> colr	montana -> montn
modeling -> modlng	furniture -> furntr
modelling -> modlng	furnature -> furntr

In the preceding example, you would most likely want to exclude **mountain** and **montana** from being grouped together. Therefore, you could enter them in the Exceptions section in the following manner:

mountain montana

Important: In some cases, fuzzy grouping exceptions do not stop 2 words from being paired because certain synonym rules are being applied. In that case, you may want to try entering synonyms using the exclamation mark wildcard (!) to prohibit the words from becoming synonymous in the output. For more information, see [Defining synonyms](#).

Formatting Rules for Fuzzy Grouping Exceptions

- Define only one exception pair per line.
- Use simple or compound words.
- Use only lowercase characters for the words. Uppercase words will be ignored.
- Use a **TAB** character to separate each word in a pair.

Related information

- [About Advanced Resources](#)
 - [Finding](#)
 - [Replacing](#)
 - [Target Language for Resources](#)
 - [Nonlinguistic Entities](#)
 - [Language Handling](#)
 - [Viewing and working in source mode](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Nonlinguistic Entities

When working with certain kinds of data, you might be very interested in extracting dates, social security numbers, percentages, or other nonlinguistic entities. These entities are explicitly declared in the configuration file, in which you can enable or disable the entities. See the topic [Configuration](#) for more information. In order to optimize the output from the extraction engine, the input from nonlinguistic processing is normalized to group like entities according to predefined formats. See the topic [Normalization](#) for more information.

Note: You can turn on and off nonlinguistic entity extraction in the extraction settings.

Available Nonlinguistic Entities

The nonlinguistic entities in the following table can be extracted. The type name is in parentheses.

Table 1. Nonlinguistic entities that can be extracted

Addresses	(<Address>)
Amino acids	(<Aminoacid>)
Currencies	(<Currency>)
Dates	(<Date>)
Delay	(<Delay>)
Digits	(<Digit>)
E-mail addresses	(<email>)
HTTP/URL addresses	(<url>)
IP address	(<IP>)
Organizations	(<Organization>)
Percentages	(<Percent>)
Products	(<Product>)
Proteins	(<Gene>)
Phone numbers	(<PhoneNumber>)
Times	(<Time>)
U.S. social security	(<SocialSecurityNumber>)
Weights and measures	(<Weights-Measures>)

Cleaning Text for Processing

Before nonlinguistic entities extraction occurs, the input text is cleaned. During this step, the following temporary changes are made so that nonlinguistic entities can be identified and extracted as such:

- Any sequence of two or more spaces is replaced by a single space.
- Tabulations are replaced by space.
- Single end-of-line characters or sequence characters are replaced by a space, while multiple end-of-line sequences are marked as end of a paragraph. End of line can be denoted by carriage returns (CR) and line feed (LF) or even both together.
- HTML and XML tags are temporarily stripped and ignored.
- [Regular Expression Definitions](#)
- [Normalization](#)
- [Configuration](#)

Related information

- [Regular Expression Definitions](#)
- [Normalization](#)
- [Configuration](#)
- [About Advanced Resources](#)
- [Finding](#)
- [Replacing](#)
- [Target Language for Resources](#)
- [Fuzzy Grouping](#)
- [Language Handling](#)
- [Viewing and working in source mode](#)
- [Microsoft Internet Explorer settings for Help](#)

Regular Expression Definitions

When extracting nonlinguistic entities, you may want to edit or add to the regular expression definitions that are used to identify regular expressions. This is done in the Regular Expression Definitions section in the Advanced Resources tab. See the topic [About Advanced Resources](#) for more information.

The file is broken up into distinct sections. The first section is called `[macros]`. In addition to that section, an additional section can exist for each nonlinguistic entity. You can add sections to this file. Within each section, rules are numbered (`regexp1`, `regexp2`, and so on). These rules

must be numbered sequentially from 1–n. Any break in numbering will cause the processing of this file to be suspended altogether.

In certain cases, an entity is language dependent. An entity is considered to be language dependent if it takes a value other than 0 for the language parameter in the configuration file. See the topic [Configuration](#) for more information. When an entity is language dependent, the language must be used to prefix the section name, such as `[english/PhoneNumber]`. That section would contain rules that apply only to English phone numbers when the `PhoneNumber` entity is given a value of 2 for the language.

Important! If you make changes to this file or any other in the editor and the extraction engine no longer works as desired, use the Reset to Original option on the toolbar to reset the file to the original shipped content. This file requires a certain level of familiarity with regular expressions. If you require additional assistance in this area, please contact IBM® Corp. for help.

Special Characters . [] {} () \ * + ? | ^ \$

All characters match themselves except for the following special characters, which are used for a specific purpose in expressions: . [{ () \ * + ? | ^ \$ To use these characters as such, they must be preceded by a backslash (\) in the definition.

For example, if you were trying to extract Web addresses, the full stop character is very important to the entity, therefore, you must backslash it such as:

```
www\.[a-z]+\.[a-z]+
```

Repetition Operators and Quantifiers ? + * {}

To enable the definitions to be more flexible, you can use several wildcards that are standard to regular expressions. They are * ? +

- Asterisk * indicates that there are *zero or more* of the preceding string. For example, `ab*c` matches "ac", "abc", "abbbc", and so on.
- Plus sign + indicates that there is *one or more* of the preceding string. For example, `ab+c` matches "abc", "abbc", "abbbc", but not "ac".
- Question mark ? indicates that there is *zero or one* of the preceding string. For example, `model?ing` matches both "modeling" and "modeling".
- Limiting repetition with brackets {} indicates the bounds of the repetition. For example, `[0-9]{n}` matches a digit repeated exactly n times. For example, `[0-9]{4}` will match "1998", but neither "33" nor "19983".

`[0-9]{n,}` matches a digit repeated *n or more* times. For example, `[0-9]{3,}` will match "199" or "1998", but not "19".

`[0-9]{n,m}` matches a digit repeated between *n and m times inclusive*. For example, `[0-9]{3,5}` will match "199", "1998" or "19983", but not "19" nor "19983".

Optional Spaces and Hyphens

In some cases, you want to include an optional space in a definition. For example, if you wanted to extract currencies such as "uruguayan pesos", "uruguayan peso", "uruguay pesos", "uruguay peso", "pesos" or "peso", you would need to deal with the fact that there may be two words separated by a space. In this case, this definition should be written as `(uruguayan |uruguay)?pesos?`. Since uruguayan or uruguay are followed by a space when used with pesos/peso, the optional space must be defined within the optional sequence `(uruguayan |uruguay)`. If the space was not in the optional sequence such as `(uruguayan|uruguay)?pesos?`, it would not match on "pesos" or "peso" since the space would be required.

If you are looking for a series of things including a hyphen characters (-) in a list, then the hyphen must be defined last. For example, if you are looking for a comma (,) or a hyphen (-), use `[,-]` and never `[-,]`.

Order of Strings in Lists and Macros

You should always define the longest sequence before a shorter one or else the longest will never be read since the match will occur on the shorter one. For example, if you were looking for strings "billion" or "bill", then "billion" must be defined before "bill". So for instance `(billion|bill)` and not `(bill|billion)`. This also applies to macros, since macros are lists of strings.

Order of Rules in the Definition Section

Define one rule per line. Within each section, rules are numbered (`regexp1`, `regexp2`, and so on). These rules must be numbered sequentially from 1–n. Any break in numbering will cause the processing of this file to be suspended altogether. To disable an entry, place a # symbol at the beginning of each line used to define the regular expression. To enable an entry, remove the # character before that line.

In each section, the most specific rules must be defined before the most general ones to ensure proper processing. For example, if you were looking for a date in the form "month year" and in the form "month", then the "month year" rule must be defined before the "month" rule. Here is how it should be defined:

```
#@# January 1932
regexp1=$ (MONTH) ,? [0-9]{4}

#@# January
regexp2=$ (MONTH)
```

and not

```
#@# January
regexp1=$ (MONTH)
```

```
#@# January 1932
regexp2=$ (MONTH) ,? [0-9]{4}
```

Using Macros in Rules

Whenever a specific sequence is used in several rules, you can use a macro. Then, if you need to change the definition of this sequence, you will need to change it only once, and not in all the rules referring to it. For example, assuming you had the following macro:

```
MONTH=( (january|february|march|april|june|july|august|september|october|
november|december) |(jan|feb|mar|apr|may|jun|jul|aug|sep|oct|nov|dec) (\.)?)
```

Whenever you refer to the name of the macro, it must be enclosed in \$(), such as: `regexp1=$ (MONTH)`

All macros must be defined in the `[macros]` section.

Related information

- [Nonlinguistic Entities](#)
 - [Normalization](#)
 - [Configuration](#)
-

Normalization

When extracting nonlinguistic entities, the entities encountered are normalized to group like entities according to predefined formats. For example, currency symbols and their equivalent in words are treated as the same. The normalization entries are stored in the Normalization section in the Advanced Resources tab. See the topic [About Advanced Resources](#) for more information. The file is broken up into distinct sections.

Important! This file is for advanced users only. It is highly unlikely that you would need to change this file. If you require additional assistance in this area, please contact IBM® Corp. for help.

Formatting Rules for Normalization

- Add only one normalization entry per line.
- Strictly respect the sections in this file. No new sections can be added.
- To disable an entry, place a # symbol at the beginning of that line. To enable an entry, remove the # character before that line.

English Dates in Normalization

By default dates in an English template are recognized in the American style date format; that is: month, date, year. If you need to change that to the day, month, year format, disable the "format:US" line (by adding # at the beginning of the line) and enable "format:UK" (by removing the # from that line).

Related information

- [Nonlinguistic Entities](#)
 - [Regular Expression Definitions](#)
 - [Configuration](#)
-

Configuration

You can enable and disable the nonlinguistic entity types that you want to extract in the nonlinguistic entity configuration file. By disabling the entities that you do not need, you can decrease the processing time required. This is done in the Configuration section in the Advanced Resources tab. See the topic [About Advanced Resources](#) for more information. If nonlinguistic extraction is enabled, the extraction engine reads this configuration file during the extraction process to determine which nonlinguistic entity types should be extracted.

The syntax for this file is as follows:

```
#name<TAB>Language<TAB>Code
```

Table 1. Syntax for configuration file

Column label	Description
#name	The wording by which nonlinguistic entities will be referenced in the two other required files for nonlinguistic entity extraction. The names used here are case sensitive.

Column label	Description
Language	The language of the documents . It is best to select the specific language; however, an Any option exists. Possible options are: 0 = Any which is used whenever a regexp is not specific to a language and could be used in several templates with different languages, for instance an IP/URL/email addresses; 1 = French; 2 = English; 4 = German; 5 = Spanish; 6 = Dutch; 8 = Portuguese; 10 = Italian.
Code	Part-of-speech code. Most entities take a value of "s" except in a few cases. Possible values are: s = stopword; a = adjective; n = noun. If enabled, nonlinguistic entities are first extracted and the extraction patterns are applied to identify its role in a larger context. For example, percentages are given a value of "a." Suppose that 30% is extracted as an nonlinguistic entity. It would be identified as an adjective. Then if your text contained "30% salary increase," the "30%" nonlinguistic entity fits the part-of-speech pattern "ann" (adjective noun noun).

Order in Defining Entities

The order in which the entities are declared in this file is important and affects how they are extracted. They are applied in the order listed. Changing the order will change the results. The most specific nonlinguistic entities must be defined before more general ones.

For example, the nonlinguistic entity "**Aminoacid**" is defined by:

```
regexp1=($ (AA) -? $ (NUM))
```

where **\$ (AA)** corresponds to "(ala|arg|asn|asp|cys|gln|glu|gly|his|ile|leu|lys|met|phe|pro|ser)", which are specific 3-letter sequences corresponding to particular amino acids.

On the other hand, the nonlinguistic entity "**Gene**" is more general and is defined by:

```
regexp1=p{0-9}{2,3}
regexp2=[a-z]{2,4}-?[0-9]{1,3}-?[r]
regexp3=[a-z]{2,4}-?[0-9]{1,3}-?p?
```

If "**Gene**" is defined before "**Aminoacid**" in the Configuration section, then "**Aminoacid**" will never be matched, since **regexp3** from "**Gene**" will always match first.

Formatting Rules for Configuration

- Use a **TAB** character to separate each entry in a column.
- Do not delete any lines.
- Respect the syntax shown in the preceding table.
- To disable an entry, place a # symbol at the beginning of that line. To enable an entity, remove the # character before that line.

Related information

- [Nonlinguistic Entities](#)
- [Regular Expression Definitions](#)
- [Normalization](#)
- [Microsoft Internet Explorer settings for Help](#)

Language Handling

Every language used today has special ways of expressing ideas, structuring sentences, and using abbreviations. In the Language Handling section, you can edit extraction patterns, force definitions for those patterns, and declare abbreviations for the language that you have selected in the Language drop-down list.

- Extraction patterns. See the topic [Extraction patterns](#) for more information.
- Forced definitions. See the topic [Forced Definitions](#) for more information.
- Abbreviations. See the topic [Abbreviations](#) for more information.
- [Extraction patterns](#)
- [Forced Definitions](#)
- [Abbreviations](#)

Related information

- [About Advanced Resources](#)
- [Finding](#)
- [Replacing](#)
- [Target Language for Resources](#)
- [Fuzzy Grouping](#)
- [Nonlinguistic Entities](#)

- [Viewing and working in source mode](#)
 - [Extraction patterns](#)
 - [Forced Definitions](#)
 - [Abbreviations](#)
-

Extraction patterns

When extracting information from your documents , the extraction engine applies a set of parts-of-speech extraction patterns to a "stack" of words in the text to identify candidate terms (words and phrases) for extraction. You can add or modify the extraction patterns.

Parts of speech include grammatical elements, such as nouns, adjectives, past participles, determiners, prepositions, coordinators, first names, initials, and particles. A series of these elements makes up a part-of-speech extraction pattern. In IBM® Corp. text mining products, each part of speech is represented by a single character to make it easier to define your patterns. For instance, an adjective is represented by the lowercase letter *a*. The set of supported codes appears by default at the top of each default extraction patterns section along with a set of patterns and examples of each pattern to help you understand each code that is used.

Formatting rules for extraction patterns

- One pattern per line.
- Use # at the beginning of a line to disable a pattern.

The order in which you list the extraction patterns is very important because a given sequence of words is read only once by the extraction engine and is assigned to the first extraction patterns for which the engine finds a match.

Supported parts of speech codes

Following is a table of all supported parts of speech codes defined in the English compiled dictionary.

All the parts of speech that are used in a particular template are listed at the top of Advanced Resources...> Extraction patterns.

The main difference between the basic resources template and the opinions template is that when minimal determiners ("d") and prepositions ("c") are used in basic, their extended equivalents ("e" and "r") are used in opinions. "0" and "1" have a limited use in all the opinions templates. See Advanced Resources...> Language Handling (English)...> Forced Definitions and Extraction patterns.

Other English templates may use some parts of speech not listed in the dictionary (for instance, "w" and "W", in the Market Intelligence template). But in that case, those parts of speech are assigned to specific words under Advanced Resources...> Forced Definitions.

Table 1. Supported parts of speech codes

Code	Meaning	Example
a	adjective	abdominal, blue...
A	unused	unused
b	adverb	frequently, often, very, ...
B	unused	unused
c	preposition	"of"
C	internal code for misspelled words	
d	determiner	"the"
D	unused	unused
e	extended determiner	the, an, my, your...
E	unused	unused
f	first name	John, Mary...
F	unused	unused
g	unused	unused
G	nationality adjective	french, american...
h	unused	unused
H	unused	unused
i	initial all single letters followed by ":"	"a.", "w." and some single letters such as "w" (used to extract person names such as John W. Doe)
I	unused	unused
j	unused	unused
J	unused	unused
k	unused	unused
K	unused	unused

Code	Meaning	Example
l	unused	unused
L	unused	unused
m	noun or unknown	dog, ibm
M	unused	unused
n	noun	dog
N	a few proper nouns	ibm
o	coordination	"and", "&"
O	unused	unused
p	past participle	abandoned, accessorized...
P	unused	unused
q	unused	unused
Q	qualifier	expensive, small, good, ...
r	extended preposition	of, among, against, from...
R	unused	unused
s	stop word	any word that we do not want to extract
S	unused	unused
t	title	mrs., mrs, captain, brig., ...
T	unused	unused
u	unknown by definition, not in dictionary	
U	unused	unused
v	verb	eat, eats, ate, eating, ...
V	infinitive verb	eat, ...
w	unused	unused
W	unused	unused
x	auxiliary	be
X	unused	unused
y	particle	von, di, de, ... (used to extract person names: John von Doe)
Y	unused	unused
z	unused	unused
Z	unused	unused
0	opinion adverb	Only in Opinions. See Advanced resources > Language Handling (English) > Forced Definitions.
1	"to" in opinions	See Advanced resources > Language Handling (English) > Forced Definitions
2	unused	unused
3	unused	unused
4	unused	unused
5	unused	unused
6	unused	unused
7	unused	unused
8	unused	unused
9	unused	unused

Forced Definitions

When extracting information from your documents , the extraction engine scans the text and identifies the part of speech for every word it encounters. In some cases, a word could fit several different roles depending on the context. If you want to force a word to take a particular part-of-speech role or to exclude the word completely from processing, you can do so in the Forced Definition section of the Advanced Resources tab. See the topic [About Advanced Resources](#) for more information.

To force a part-of-speech role for a given word, you must add a line to this section using the following syntax:

term: code

Table 1. Syntax description

Entry	Description
term	A term name.

Entry	Description
code	A single-character code representing the part-of-speech role. You can list up to six different part-of-speech codes per uniterm. Additionally, you can stop a word from being extracted into compound words/phrases by using the lowercase code s , such as additional:s .

Formatting Rules for Forced Definitions

- One line per word.
- Terms cannot contain a colon.
- Use the lowercase **s** as a part-of-speech code to stop a word from being extracted altogether.
- Use up to six part-of-speech codes per line. Supported part-of-speech codes are shown in the Extraction Patterns section. See the topic [Extraction patterns](#) for more information.
- Use the asterisk character (*) as a wildcard at the end of a string for partial matches. For example, if you enter **add*:s**, words such as **add**, **additional**, **additionally**, **addendum**, and **additive** are never extracted as a term or as part of a compound word term. However, if a word match is explicitly declared as a term in a compiled dictionary or in the forced definitions, it will still be extracted. For example, if you enter both **add*:s** and **addendum:n**, **addendum** will still be extracted if found in the text.

Related information

- [Language Handling](#)
 - [Extraction patterns](#)
 - [Abbreviations](#)
-

Abbreviations

When the extraction engine is processing text, it will generally consider any period it finds as an indication that a sentence has ended. This is typically correct; however, this handling of period characters does not apply when abbreviations are contained in the text.

If you extract terms from your text and find that certain abbreviations were mishandled, you should explicitly declare that abbreviation in this section.

Note: If the abbreviation already appears in a synonym definition or is defined as a term in a type dictionary, there is no need to add the abbreviation entry here.

Formatting Rules for Abbreviations

- Define one abbreviation per line.

Related information

- [Language Handling](#)
 - [Extraction patterns](#)
 - [Forced Definitions](#)
-

About Text Link Rules

Text link analysis (TLA) is a pattern matching technology that is used to extract relationships found in your text using a set of rules. When text link analysis is enabled for extraction, the text data is compared against these rules. When a match is found, the text link analysis pattern is extracted and presented. These rules are defined in the Text Link Rules tab.

For example, extracting concepts representing simple ideas about an organization may not be interesting enough to you, but by using TLA, you could also learn about the links between different organizations or the people associated with the organization. TLA can also be used to extract opinions about topics such as how people feel about a given product or experience.

To benefit from TLA, you must have resources that contain text link (TLA) rules. When you select a template, you can see which templates have TLA rules by whether or not they have an icon in the TLA column.

Text link analysis patterns are found in the text data during the pattern matching phase of the extraction process. During this phase, rules are compared to the text data and when a match is found, this information is extracted as a pattern. There are times when you might want to get more from text link analysis or change how something is matched. In these cases, you can refine the rules to adapt them to your particular needs. This is performed in the Text Link Rules tab.

Note: Starting with version 18.2, Type Reassignment Rules (TRRs) are available. TRRs transform a sequence of types, macros, and/or tokens into a new concept with a specific type. They can be used in Opinions templates to catch opinions with a change in polarity. For more information, see

Type Reassignment Rules.

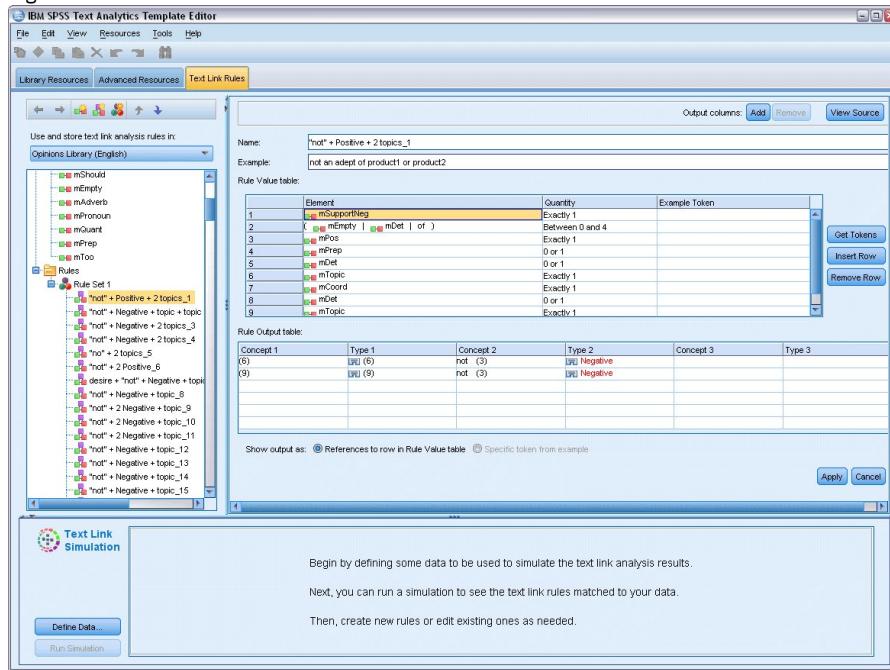
- [Where to work on text link rules](#)
- [Where to Begin](#)
- [When to Edit or Create Rules](#)
- [Simulating Text Link Analysis Results](#)
- [Navigating Rules and Macros in the Tree](#)
- [Working with Macros](#)
- [Working with Text Link Rules](#)
- [Processing Order for Rules](#)
- [Working with Rule Sets \(Multiple Pass\)](#)
- [Supported Elements for Rules and Macros](#)
- [Viewing and working in source mode](#)

Where to work on text link rules

You can edit and create rules directly in the Text Link Rules tab in the Template Editor or Resource Editor view. To help you see how rules might match text, you can run a simulation in this tab. During simulation, an extraction is run only on the sample simulation data and the text link rules are applied to see if any patterns match. Any rules that match the text are then shown in the simulation pane. Based on the matches, you can choose to edit rules and macros to change how the text is matched.

Unlike the other advanced resources, TLA rules are library-specific; therefore, you can only use the TLA rules from one library at a time. From within the Template Editor or Resource Editor, go to the Text Link Rules tab. In this tab, you can specify the library in your template that contains the TLA rules you want to use or edit. For this reason, we strongly recommend that you store all your rules in one library unless there is a very specific reason this isn't desired.

Figure 1. Text Link Rules tab



Where to Begin

There are a number of ways to start working in the Text Link Rules tab editor:

- Start by simulating results with some sample text and edit or create matching rules based on how the current set of rules extract patterns from the simulation data. See the topic [Simulating Text Link Analysis Results](#) for more information.
- Create a new rule from scratch or edit an existing rule. See the topic [Working with Text Link Rules](#) for more information.
- Work in source view directly. See the topic [Viewing and working in source mode](#) for more information.

Related information

- [About Text Link Rules](#)
 - [Where to work on text link rules](#)
 - [When to Edit or Create Rules](#)
 - [Simulating Text Link Analysis Results](#)
 - [Navigating Rules and Macros in the Tree](#)
 - [Working with Macros](#)
 - [Creating and Editing Macros](#)
 - [Disabling and Deleting Macros](#)
 - [Checking for Errors, Saving, and Cancelling](#)
 - [Special Macros: mTopic, mNonLingEntities, SEP](#)
 - [Working with Text Link Rules](#)
 - [Working with Rule Sets \(Multiple Pass\)](#)
 - [Supported Elements for Rules and Macros](#)
 - [Viewing and working in source mode](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

When to Edit or Create Rules

While the text link analysis rules delivered with each template are often adequate for extracting many simple or complex relationships from your text, there are times that you may want to make some changes to these rules or create some rules of your own. For example:

- To capture an idea or relation that isn't being extracted with the existing rules by creating a new rule or macro.
- To change the default behavior of a type you added to the resources. This usually requires you to edit a macro such as `mTopic` or `mNonLingEntities`. See the topic [Special Macros: mTopic, mNonLingEntities, SEP](#) for more information.
- To add new types to existing text link analysis rules and macros. For example, if you think the type `<Organization>` is too broad, you could create new types for organizations in several different business sectors such as `<Pharmaceuticals>`, `<Car Manufacturing>`, `<Finance>`, and so on. In this case, you must edit the text link analysis rules and/or create a macro to take these new types into account and process them accordingly.
- To add types to an existing text link analysis rule. For example, let's say you have a rule that captures the following text `john doe called jane doe` but you want this rule that captures phone communications to also capture email exchanges. You could add the nonlinguistic entity type for email to the rule so it would also capture text such as: `johndoe@ibm.com emailed janedoe@ibm.com`.
- To slightly modify an existing rule, instead of creating a new one. For example, let's say you have a rule that matches the following text `xyz is very good` but you want this rule to also capture `xyz is very, very good`.

Related information

- [About Text Link Rules](#)
 - [Where to work on text link rules](#)
 - [Where to Begin](#)
 - [Simulating Text Link Analysis Results](#)
 - [Navigating Rules and Macros in the Tree](#)
 - [Working with Macros](#)
 - [Creating and Editing Macros](#)
 - [Disabling and Deleting Macros](#)
 - [Checking for Errors, Saving, and Cancelling](#)
 - [Special Macros: mTopic, mNonLingEntities, SEP](#)
 - [Working with Text Link Rules](#)
 - [Working with Rule Sets \(Multiple Pass\)](#)
 - [Supported Elements for Rules and Macros](#)
 - [Viewing and working in source mode](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Simulating Text Link Analysis Results

In order to help define new text link rules or help understand how certain sentences are matched during text link analysis, it is often useful to take a sample piece of text and run a simulation. During simulation, an extraction is run only on the sample simulation data using the current set of linguistic resources and the current extraction settings. The goal is to obtain the simulated results and use these results to improve your rules, create new ones, or better understand how matching occurs. For each piece of text (sentence, word, or clause depending on the context), a simulation output displays the collection of tokens and any TLA rules that uncovered a pattern in that text. A **token** is defined as any word or word phrase identified during the extraction process.

Unlike the other advanced resources, TLA rules are library-specific; therefore, you can only use the TLA rules from one library at a time. From within the Template Editor or Resource Editor, go to the Text Link Rules tab. In this tab, you can specify the library in your template that contains

the TLA rules you want to use or edit. For this reason, we strongly recommend that you store all your rules in one library unless there is a very specific reason this isn't desired.

Important! We strongly recommend that if you use a data file, please ensure that the text it contains is short in order to minimize processing time. The goal of simulation is to see how a piece of text is interpreted and to understand how rules match this text. This information will help you write and edit your rules. Use the text link analysis node or run a stream with interactive session with TLA extraction enabled to obtain results for a more complete data set. This simulation is for testing and rule authoring purposes only.

- [Defining Data for Simulation](#)
- [Understanding Simulation Results](#)

Related information

- [About Text Link Rules](#)
- [Where to work on text link rules](#)
- [Where to Begin](#)
- [When to Edit or Create Rules](#)
- [Navigating Rules and Macros in the Tree](#)
- [Working with Macros](#)
- [Creating and Editing Macros](#)
- [Disabling and Deleting Macros](#)
- [Checking for Errors, Saving, and Cancelling](#)
- [Special Macros: mTopic, mNonLingEntities, SEP](#)
- [Working with Text Link Rules](#)
- [Working with Rule Sets \(Multiple Pass\)](#)
- [Supported Elements for Rules and Macros](#)
- [Viewing and working in source mode](#)
- [Defining Data for Simulation](#)
- [Understanding Simulation Results](#)
- [Microsoft Internet Explorer settings for Help](#)

Defining Data for Simulation

To help you see how rules might match text, you can run a simulation using sample data. The first step is to define the data.

Defining Data

1. Click Define Data in the simulation pane in bottom of the Text Link Rules tab. Alternatively, if no data have been previously defined, choose Tools > Run Simulation from the menus. The Simulation Data wizard opens.
2. Specify the data type by selecting one of the following:
 - Paste or enter text directly A text box is provided for you to paste some text from the clipboard or to manually enter the desired text to be processed. You can enter one sentence per line or use punctuation to break up the sentence such as periods or commas. Once you have entered your text, you can begin the simulation by clicking Run Simulation.
 - Specify a file data source This option indicates that you want to process a file that contains text. Click Next to proceed to the wizard step in which you can define the file to be processed. Once the file has been selected, you can begin the simulation by clicking Run Simulation. The following file types are supported: .txt and .text. The data file you choose is read 'as-is' during the simulation. The entire file is treated in the same manner as if you had connected a File List node to a Text Mining Node.
- Important: We strongly recommend that if you use a data file, ensure that the text it contains is short in order to minimize processing time. The goal of simulation is to see how a piece of text is interpreted and to understand how rules match this text. This information will help you write and edit your rules. Use the text link analysis node or run a stream with interactive session with TLA extraction enabled to obtain results for a more complete data set. This simulation is for testing and rule authoring purposes only.
3. To begin the simulation process, click Run Simulation. A progress dialog appears. If you are in an interactive session, the extraction settings used during simulation are those currently selected in the interactive session (see Tools > Extraction Settings in the Concepts and Categories view). If you are in the Template Editor, the extraction settings used during simulation are the default extraction settings, which are the same as those shown in the Expert tab of a Text Link Analysis node. For more information, see [Understanding Simulation Results](#).

Related information

- [Simulating Text Link Analysis Results](#)
- [Understanding Simulation Results](#)

Understanding Simulation Results

To help you see how rules might match text, you can run a simulation using sample data and review the results. From there you can change your set of rules to better fit your data. When the extraction and simulation process has completed, you will be presented with the results of the simulation.

For each "sentence" identified during extraction, you are presented with several pieces of information including the exact "sentence", the breakdown of the tokens found in this input text sentence, and finally any rules that matched text in that sentence. By "sentence", we mean either a word, sentence, or clause depending on how the extractor broke down the text into readable chunks.

A **token** is defined as any word or word phrase identified during the extraction process. For example, in the sentence *My uncle lives in New York*, the following tokens might be found during extraction: *my*, *uncle*, *lives*, *in*, and *new york*. Additionally, *uncle* could be extracted as a concept and typed as **<Unknown>**, and *new york* could also be extracted as a concept and typed as **<Location>**. All concepts are tokens but not all tokens are concepts. Tokens can also be other macros, literal strings, and word gaps. Only those words or word phrases that are typed can be concepts.

When you are working in the interactive session or resource editor, you are working at the concept level. TLA rules are more granular, and individual tokens in a sentence can be used in the definition of a rule even if they are never extracted and typed. Being able to use tokens which are not concepts offers rules even more flexibility in capturing complex relationships in your text.

If you have more than one sentence in your simulation data, you can move forward and backward through the results by clicking Next and Previous.

In those cases where a sentence does not match any TLA rule in the selected library (see library name above tree in this tab), the results are considered unmatched and the buttons Next Unmatched and Previous Unmatched are enabled to let you know that there is text for which no rule found a match and to allow you to navigate to these instances quickly.

After creating new rules, editing your rules, or changing your resources or extraction settings, you may want to rerun a simulation. To re-run a simulation, click Run Simulation in the simulation pane and the same input data will be used again.

The following fields and tables are shown in the simulation results:

Input text. The actual 'sentence' identified by the extraction process from the simulation data you defined in the wizard. By sentence, we mean either a word, sentence, or clause depending on how the extractor broken down the text into readable chunks.

System View. A collection of tokens that the extraction process has identified.

- **Input Text Token.** Each token found in the input text. Tokens were defined earlier in this topic.
- **Typed As.** If a token was identified as a concept and typed, then the associated type name (such as **<Unknown>**, **<Person>**, **<Location>**) is shown in this column.
- **Matching Macro.** If a token matched an existing macro, then the associated macro name is displayed in this column.

Rules Matched to Input Text. This table shows you any TLA rules that were matched against the input text. For each matched rule, you will see the name of the rule in the Rule Output column and the associated output values for that rule (Concept + Type pairs). You can double-click on the matched rule name to open the rule in the editor pane above the simulation pane.

Generate Rule button. If you click this button in the simulation pane, a new rule will open in the rule editor pane above the simulation pane. It will take the input text as its example. Likewise, any token that was typed or matched to a macro during simulation is automatically inserted in the Elements column in the Rule Values table. If a token was typed and matched a macro, the macro value is the one that will be used in the rule so as to simplify the rule. For example, the sentence "*I like pizza*" could be typed during simulation as **<Unknown>** and matched to macro **mTopic** if you were using the Basic English resources. In this case **mTopic** will be used as the element in the generated rule. See the topic [Working with Text Link Rules](#) for more information.

Related information

- [Simulating Text Link Analysis Results](#)
 - [Defining Data for Simulation](#)
-

Navigating Rules and Macros in the Tree

When text link analysis is performed during extraction, the text link rules stored in the library selected in the Text Link Rules tab will be used.

Unlike the other advanced resources, TLA rules are library-specific; therefore, you can only use the TLA rules from one library at a time. From within the Template Editor or Resource Editor, go to the Text Link Rules tab. In this tab, you can specify the library in your template that contains the TLA rules you want to use or edit. For this reason, we strongly recommend that you store all your rules in one library unless there is a strong or specific reason this isn't desired.

You can specify in which library you want to work in the Text Link Rules tab by selecting that library in the Use and store text link analysis rules in dropdown list in this tab. When text link analysis is performed during extraction, the text link rules stored in the library selected in the Text Link Rules tab will be used. Therefore, if you defined text link rules (TLA rules) in more than one library, only the first library in which TLA rules are found will be used for text link analysis. For this reason, we strongly recommend that you store all your rules in one library unless there is a very specific reason this isn't desired.

When you select a macro or rule in the tree, its contents are displayed in the editor pane to the right. If you right-click on any item in the tree, a context menu will open to show you what other tasks are possible, such as:

- Create a new macro in the tree and open it in the editor to the right.
- Create a new rule in the tree and open it in the editor to the right.
- Create a new rule set in the tree.
- Cut, copy, and paste items to simplify editing.
- Delete macros, rules, and rule sets to remove them from the resources.
- Disable macros, rules, and rule sets to indicate that they should be ignored during processing.
- Move rules up or down to affect processing order.

Warnings in the tree

Warnings are displayed with a yellow triangle in the tree and are there to inform you that there may be a problem. Hover the mouse pointer over the faulty macro or rule to display a pop-up explanation. In most cases, you will see something such as: Warning: No example provided; Enter an example so you need to enter an example.

If you're missing an example, or if the example doesn't match the rule, you will not be able to use the Get Tokens feature so we recommend you enter just one example per rule.

When the rule is highlighted in yellow it means that a type or macro is unknown to the TLA editor. The message will be similar to: Warning: Unknown type or macro. This is to inform you that an item that would be defined by \$something in the source view, for instance \$myType, is not a legacy type in your library, nor is it a macro.

To update the syntax checker you need to switch to another rule or macro; there is no need to recompile anything. So, for example, if rule A displays a warning because the example is missing, you need to add an example, click on either an upper or lower rule, and then go back to rule A to check that it is now correct.

Related information

- [About Text Link Rules](#)
 - [Where to work on text link rules](#)
 - [Where to Begin](#)
 - [When to Edit or Create Rules](#)
 - [Simulating Text Link Analysis Results](#)
 - [Working with Macros](#)
 - [Creating and Editing Macros](#)
 - [Disabling and Deleting Macros](#)
 - [Checking for Errors, Saving, and Cancelling](#)
 - [Special Macros: mTopic, mNonLingEntities, SEP](#)
 - [Working with Text Link Rules](#)
 - [Working with Rule Sets \(Multiple Pass\)](#)
 - [Supported Elements for Rules and Macros](#)
 - [Viewing and working in source mode](#)
 - [Microsoft Internet Explorer settings for Help](#)
-

Working with Macros

Macros can simplify the appearance of text link analysis rules by allowing you to group types, other macros, and literal (word) strings together with an OR operator (|). The advantage to using macros is that not only can you reuse macros in multiple text link analysis rules to simplify them, but it also enables you to make updates in one macro rather than having to make updates throughout all of your text link analysis rules. Most shipped TLA rules contain predefined macros. Macros appear at the top of the tree in the leftmost pane of the Text Link Rules tab.

The following fields and tables are shown in the simulation results:

Name. A unique name identifying this macro. We recommend that you prefix macro names with a lowercase m to help you identify macros quickly in your rules. When you manually refer to macros in your rules (by inline editing or in the source view) you have to use the \$ character prefix so that the extraction process knows to look for this special name. However, if you drag and drop the macro name or add it through the context menus, the product will automatically recognize it as a macro and no \$ will be added.

Macro Value table.

- A number of rows representing all of the possible values this macro can represent. These values are case-sensitive.
- These values can include one or a combination of types, literal strings, word gaps, or macros. See the topic [Supported Elements for Rules and Macros](#) for more information.
- To enter a value for an element in a macro, double-click the row you want to work in. An editable text box appears in which you can enter a type reference, a macro reference, a literal string, or a word gap. Alternatively, right-click in the cell to display a contextual menu offering lists of common macros, type names, and nonlinguistic type names. To reference a type or a macro you must precede the macro or type

name with a '\$' character such as \$mTopic for the macro mTopic. When combining arguments, you must use parentheses () to group the arguments and the character | to indicate a Boolean OR.

- You can add or remove rows in the Macro Value table using the buttons to its right.
- Enter each element in its own row. For example, if you wanted to create a macro that represents one of 3 literal strings such as am OR was OR is, you would enter each literal string on a separate row in the view, and your Macro table would contain 3 rows.

- [Creating and Editing Macros](#)
- [Disabling and Deleting Macros](#)
- [Checking for Errors, Saving, and Cancelling](#)
- [Special Macros: mTopic, mNonLingEntities, SEP](#)

Related information

- [About Text Link Rules](#)
- [Where to work on text link rules](#)
- [Where to Begin](#)
- [When to Edit or Create Rules](#)
- [Simulating Text Link Analysis Results](#)
- [Navigating Rules and Macros in the Tree](#)
- [Creating and Editing Macros](#)
- [Disabling and Deleting Macros](#)
- [Checking for Errors, Saving, and Cancelling](#)
- [Special Macros: mTopic, mNonLingEntities, SEP](#)
- [Working with Text Link Rules](#)
- [Working with Rule Sets \(Multiple Pass\)](#)
- [Supported Elements for Rules and Macros](#)
- [Viewing and working in source mode](#)
- [Creating and Editing Rules](#)
- [Processing Order for Rules](#)

Creating and Editing Macros

You can create new macros or edit existing ones. Follow the guidelines and descriptions for the macro editor. See the topic [Working with Macros](#) for more information.

Creating New Macros

1. From the menus, choose Tools > New Macro. Alternatively, click the New Macro icon in the tree toolbar to open a new macro in the editor.
2. Enter a unique name and define the macro value elements.
3. Click Apply when finished to check for errors.

Editing Macros

1. Click the macro name in the tree. The macro opens in the editor pane on the right.
2. Make your changes.
3. Click Apply when finished to check for errors.

Related information

- [About Text Link Rules](#)
- [Where to work on text link rules](#)
- [Where to Begin](#)
- [When to Edit or Create Rules](#)
- [Simulating Text Link Analysis Results](#)
- [Navigating Rules and Macros in the Tree](#)
- [Working with Macros](#)
- [Disabling and Deleting Macros](#)
- [Checking for Errors, Saving, and Cancelling](#)
- [Special Macros: mTopic, mNonLingEntities, SEP](#)
- [Working with Text Link Rules](#)
- [Working with Rule Sets \(Multiple Pass\)](#)
- [Supported Elements for Rules and Macros](#)
- [Viewing and working in source mode](#)
- [Creating and Editing Rules](#)
- [Processing Order for Rules](#)
- [Disabling and Deleting Rules](#)

- [Checking for Errors, Saving, and Cancelling](#)
-

Disabling and Deleting Macros

Disabling Macros

If you want a macro to be ignored during processing, you can disable it. Doing so may cause warnings or errors in any rules that still reference this disabled macro. Take caution when deleting and disabling macros.

1. Click the macro name in the tree. The macro opens in the editor pane on the right.
2. Right-click on the name.
3. From the context menus, choose Disable. The macro icon becomes gray and the macro itself becomes uneditable.

Deleting Macros

If you want to get rid of a macro, you can delete it. Doing so may cause errors in any rules that still reference this macro. Take caution when deleting and disabling macros.

1. Click the macro name in the tree. The macro opens in the editor pane on the right.
2. Right-click on the name.
3. From the context menus, choose Delete. The macro disappears from the list.

Related information

- [About Text Link Rules](#)
 - [Where to work on text link rules](#)
 - [Where to Begin](#)
 - [When to Edit or Create Rules](#)
 - [Simulating Text Link Analysis Results](#)
 - [Navigating Rules and Macros in the Tree](#)
 - [Working with Macros](#)
 - [Creating and Editing Macros](#)
 - [Checking for Errors, Saving, and Cancelling](#)
 - [Special Macros: mTopic, mNonLingEntities, SEP](#)
 - [Working with Text Link Rules](#)
 - [Working with Rule Sets \(Multiple Pass\)](#)
 - [Supported Elements for Rules and Macros](#)
 - [Viewing and working in source mode](#)
 - [Creating and Editing Rules](#)
 - [Processing Order for Rules](#)
-

Checking for Errors, Saving, and Cancelling

Applying Macro Changes

If you click outside of the macro editor or if you click Apply, the macro is automatically scanned for errors. If an error is found, you will need to fix it before moving on to another part of the application.

However, if less serious errors are detected, only a warning is given. For example, if your macro contains incomplete or unreferenced definitions to types or other macros, a warning message is displayed. Once you click Apply, any uncorrected warnings cause a warning icon to appear to the left of the macro name in the Rules and Macro Tree in the left pane.

Applying a macro does not mean that your macro is permanently saved. Applying will cause the validation process to check for errors and warnings.

Saving Resources inside an Interactive Workbench Session

1. To save the changes you made to your resources during an interactive workbench session so you can get them next time you run your stream, you must:
 - Update your modeling node to make sure that you can get these same resources next time you execute your stream. See the topic [Updating Modeling Nodes and Saving](#) for more information. Then save your stream. To save your stream, do so in the main IBM® SPSS® Modeler window after updating the modeling node.
2. To save the changes you made to your resources during an interactive workbench session so that you can use them in other streams, you can:

- Update the template you used or make a new one. See the topic [Making and Updating Templates](#) for more information. This will not save the changes for the current node (see previous step)
- Or, update the TAP you used. See the topic [Updating Text Analysis Packages](#) for more information.

Saving Resources inside the Template Editor

1. First, publish the library. See the topic [Publishing Libraries](#) for more information.
2. Then, save the template through File > Save Resource Template in the menus.

Cancelling Macro Changes

1. If you wish to discard the changes, click Cancel.

Related information

- [About Text Link Rules](#)
- [Where to work on text link rules](#)
- [Where to Begin](#)
- [When to Edit or Create Rules](#)
- [Simulating Text Link Analysis Results](#)
- [Navigating Rules and Macros in the Tree](#)
- [Working with Macros](#)
- [Creating and Editing Macros](#)
- [Disabling and Deleting Macros](#)
- [Special Macros: mTopic, mNonLingEntities, SEP](#)
- [Working with Text Link Rules](#)
- [Working with Rule Sets \(Multiple Pass\)](#)
- [Supported Elements for Rules and Macros](#)
- [Viewing and working in source mode](#)
- [Creating and Editing Rules](#)
- [Processing Order for Rules](#)

Special Macros: mTopic, mNonLingEntities, SEP

The Opinions template (and like templates) as well as the Basic Resources templates are shipped with two special macros called **mTopic** and **mNonLingEntities**.

mTopic

By default, the macro **mTopic** groups all the types shipped in the template that are likely to be connected with an opinion, such as the following Core library types: **<Person>**, **<Organization>**, **<Location>**, and so on, as long as the type is not an opinion type (for example, **<Negative>** or **<Positive>**) or a type defined as a nonlinguistic entity in the Advanced Resources.

Whenever you create a new type in an Opinions (or similar) template, the product assumes that unless this type is specified in another macro or in the nonlinguistic entities section of the Advanced Resource tab, it will be treated the same way as the other types defined in the macro **mTopic**.

Let's say you created new types in the resources from an Opinions template: **<Vegetables>** and **<Fruit>**. Without having to make any changes, your new types are treated as **mTopic** types so you can automatically uncover the positive, negative, neutral, and contextual opinions about your new types. During extraction, for example, the sentence "*I enjoy broccoli, but I hate grapefruit*" would produce the following 2 output patterns:

```
broccoli <Vegetables> + like <Positive>
grapefruit <Fruit> + dislike <Negative>
```

However, if you want to process those types differently than the other types in **mTopic**, you can either add the type name to an existing macro such as **mPos**, which groups all positive opinion types, or create a new macro that you can later reference in one or more rules.

Important! If you create a new type such as **<Vegetables>**, this new type will be included as a type in **mTopic**, however, this type name will not be explicitly visible in the macro definition.

mNonLingEntities

Similarly, if you add new nonlinguistic entities in the Nonlinguistic Entities section of the Advanced Resources tab, they will be automatically processed as **mNonLingEntities** unless specified otherwise. See the topic [Nonlinguistic Entities](#) for more information.

SEP

Related information

- [About Text Link Rules](#)
 - [Where to work on text link rules](#)
 - [Where to Begin](#)
 - [When to Edit or Create Rules](#)
 - [Simulating Text Link Analysis Results](#)
 - [Navigating Rules and Macros in the Tree](#)
 - [Working with Macros](#)
 - [Creating and Editing Macros](#)
 - [Disabling and Deleting Macros](#)
 - [Checking for Errors, Saving, and Cancelling](#)
 - [Working with Text Link Rules](#)
 - [Working with Rule Sets \(Multiple Pass\)](#)
 - [Supported Elements for Rules and Macros](#)
 - [Viewing and working in source mode](#)
 - [Creating and Editing Rules](#)
 - [Processing Order for Rules](#)
-

Working with Text Link Rules

A text link analysis rule is a Boolean query that is used to perform a match on a sentence. Text link analysis rules contain one or more of the following arguments: types, macros, literal strings, or word gaps. You must have at least one text link analysis rule in order to extract TLA results.

The following areas and fields are displayed in the Text Link Rules tab, Rule Editor:

Name field. The unique name for the text link rule.

Example field. Optionally, you can include an example sentence or word sequence that would be captured by this rule. We recommend using examples. In this editor, you will be able to generate tokens from this example text to see how it matches the rule and how it will be output. A **token** is defined as any word or word phrase identified during the extraction process. For example, in the sentence *My uncle lives in New York*, the following tokens might be found during extraction: *my*, *uncle*, *lives*, *in*, and *new york*. Additionally, *uncle* could be extracted as a concept and typed as **<Unknown>**, and *new york* could also be extracted as a concept and typed as **<Location>**. All concepts are tokens but not all tokens are concepts. Tokens can also be other macros, literal strings, and word gaps. Only those words or word phrases that are typed can be concepts.

Rule Value table. This table contains the elements of the rule that are used for matching a rule to a sentence. You can add or remove rows in the table using the buttons to its right. The table consists of 3 columns:

- Element column. Enter values as one or a combination of types, literal strings, word gaps (**<Any Token>**), or macros. See the topic [Supported Elements for Rules and Macros](#) for more information. Double-click the element cell to enter the information directly. Alternatively, right-click in the cell to display a contextual menu offering lists of common macros, type names, and nonlinguistic type names. Keep in mind that if you enter the information into the cell by typing it in, precede the macro or type name with a '\$' character such as **\$mTopic** for the macro **mTopic**. The order in which you create your element rows is critical to how the rule will be matched to the text. When combining arguments, you must use parentheses () to group the arguments and the character | to indicate a Boolean OR. Keep in mind that values are case-sensitive.
- Quantity column. This indicates the minimum and maximum number of times the element must be found for a match to occur. For example, if you want to define a gap, or a series of words, between two other elements of anywhere from 0 to 3 words, you could choose Between 0 and 3 from the list or enter the numbers directly into the dialog box. The default is 'Exactly 1'. In some cases you will want to make an element optional. If this is the case, then it will have a minimum quantity of 0 and a maximum quantity greater than 0 (i.e. 0 or 1, between 0 and 2). Note that the first element in a rule cannot be optional, meaning it cannot have a quantity of 0.
- Example Token column. If you click Get Tokens, the program breaks the Example text down into tokens and uses those tokens to fill this column with those that match the elements you defined. You can also see these tokens in the output table if you choose to.

Rule Output table Each row in this table defines how the TLA pattern output will appear in the results. Rule output can produce patterns of up to six Concept/Type column pairs, each representing a *slot*. For example, the type pattern **<Location> + <Positive>** is a two slot pattern meaning that it is made up of 2 Concept/Type column pairs.

Note: Terms in the Element column of the Rule Value table, or in any of the Concept columns of the Rule Output table cannot start with any of the following characters: ` , #, %, ^, *, _, :, <, >, /, \, or ".

Just as language gives us the freedom to express the same basic ideas in many different ways, so you might have a number of rules defined to capture the same basic idea. For example, the text "*Paris is a place I love*" and the text "*I really, really like Paris and Florence*" represent the same basic idea -- that Paris is liked -- but are expressed differently and would require two different rules to both be captured. However, it is easier to work with the pattern results if similar ideas are grouped together. For this reason, while you might have 2 different rules to capture these 2 phrases, you could define the same output for both rules, such as the type pattern **<Location> + <Positive>** so that it represents both texts. And in this way, you can see that the output does not always mimic the structure or order of the words found in the original text.

Furthermore, such a type pattern could match other phrases and could produce concept patterns such as: `paris + like` and `tokyo + like`.

To help you define the output quickly with fewer errors, you can use the context menu to choose the element you want to see in the output. Alternatively, you can also drag and drop elements from the Rule Value table into the output. For example, if you have a rule that contains a reference to the `mTopic` macro in row 2 of the Rule Value table, and you want that value to be in your output, you can simply drag/drop the element for `mTopic` to the first column pair in the Rule Output table. Doing so will automatically populate both the Concept and Type for the pair you've selected. Or if you want the output to begin with the type defined by the third element (row 3) of the rule value table, then drag that type from the Rule Value table to the Type 1 cell in the output table. The table will update to show the row reference in parenthesis (3).

Alternatively, you can enter these references manually into the table by double-clicking the cell in each Concept column you want to output and entering the `$` symbol followed by the row number, such as `$2` to refer to the element defined in row 2 of the Rule Value table. When you enter the information manually, you need to also define the Type column, enter the `#` symbol followed by the row number, such as `#2` to refer to the element defined in row 2 of the Rule Value table.

Furthermore, you might even combine methods. Let's say you had the type `<Positive>` in row 4 of your Rule Value table. You could drag it to the `Type` 2 column and then double-click the cell in the `Concept` 2 column and then manually enter the word `'not'` in front of it. The output column would then read `not (4)` in the table, or if you were in the edit mode or source mode `not $4`. Then you could right-click in the Type 1 column and select, for example, the macro called `mTopic`. Then this output could result in a concept pattern such as: `car + bad`.

Most rules have only one output row but there are times when more than one output is possible and desired. In this case, define one output per row in the Rule Output table.

Important: Keep in mind that other linguistic handling operations are performed during the extraction of TLA patterns. So when the output reads `t$3\t#3`, this means that the pattern will ultimately display the final concept for the third element and the final type for the third element after all linguistic processing is applied (synonyms and other groupings).

- Show output as. By default, the option References to row in Rule Value table is selected and the output is shown by using the numerical references to the row as defined in the Rule Value tab. If you previously clicked Get Tokens and have tokens in the Example Tokens column in the Rule Value table, you can choose to see the output for these specific tokens by choosing the option .

Note: If there are not enough concept/type output pairs shown in the output table, you can add another pair by clicking the Add button in the editor toolbar. If 3 pairs are currently shown and you click add, 2 more columns (Concept 4 and Type 4) are added to the table. This means that you will now see 4 pairs in the output table for all rules. You can also remove unused pairs as long as no other rule in the set of rules in this library uses that pair.

Example Rule

Let's suppose your resources contain the following text link analysis rule and that you have enabled the extraction of TLA results:

Figure 1. Text Link Rules tab: Rule Editor

Element	Quantity	Example Token
1 mSupportNeg	Exactly 1	isn't
2	0 or 1	
3 (anything ((any a one) thing. ?))	Exactly 1	anything
4	Between 0 and 2	that i
5 mNeg	Exactly 1	disliked
6 (about with in)	Exactly 1	about
7	0 or 1	
8 mDet	0 or 1	the

Concept 1	Type 1	Concept 2	Type 2	Concept 3	Type 3
product (9)	Products (9)	no dislike (5)	Positive		

Show output as: References to row in Rule Value table Specific token from example

Apply Cancel

Whenever you extract, the extraction engine will read each sentence and will try to match the following sequence:

Table 1. Extraction sequence example

Element (row)	Description of the arguments
1	The concept from one of the types represented by the macros <code>mPos</code> or <code>mNeg</code> or from the type <code><Uncertain></code> .
2	A concept typed as one of the types represented by the macro <code>mTopic</code> .

Element (row)	Description of the arguments
3	One of the words represented by the macro mBe .
4	An optional element, 0 or 1 words, also referred to as a word gap or <Any Token>
5	A concept typed as one of the types represented by the macro mTopic .

The output table shows that all that is wanted from this rule is a pattern where any concept or type corresponding to the **mTopic** macro that was defined in row 5 in the Rule Value table + any concept or type corresponding to the **mPos**, **mNeg**, or **<Uncertain>** as was defined in row 1 in the Rule Value table. This could be **sausage + like** or **<Unknown> + <Positive>**.

- [Creating and Editing Rules](#)
- [Disabling and Deleting Rules](#)
- [Checking for Errors, Saving, and Cancelling](#)

Creating and Editing Rules

You can create new rules or edit existing ones. Follow the guidelines and descriptions for the rule editor. See the topic [Working with Text Link Rules](#) for more information.

Creating New Rules

1. From the menus, choose Tools > New Rule. Alternatively, click the New rule icon in the tree toolbar to open a new rule in the editor.
2. Enter a unique name and define the rule value elements.
3. Click Apply when finished to check for errors.

Editing Rules

1. Click the rule name in the tree. The rule opens in the editor pane on the right.
2. Make your changes.
3. Click Apply when finished to check for errors.

Related information

- [Creating and Editing Macros](#)
- [Working with Text Link Rules](#)
- [Disabling and Deleting Rules](#)
- [Checking for Errors, Saving, and Cancelling](#)
- [Processing Order for Rules](#)
- [Working with Macros](#)
- [Disabling and Deleting Macros](#)
- [Checking for Errors, Saving, and Cancelling](#)
- [Special Macros: mTopic, mNonLingEntities, SEP](#)

Disabling and Deleting Rules

Disabling Rules

If you want a rule to be ignored during processing, you can disable it. Take caution when deleting and disabling rules.

1. Click the rule name in the tree. The rule opens in the editor pane on the right.
2. Right-click on the name.
3. From the context menus, choose Disable. The rule icon becomes gray and the rule itself becomes uneditable.

Deleting Rules

If you want to get rid of a rule, you can delete it. Take caution when deleting and disabling rules.

1. Click the rule name in the tree. The rule opens in the editor pane on the right.
2. Right-click on the name.
3. From the context menus, choose Delete. The rule disappears from the list.

Related information

- [Creating and Editing Macros](#)
- [Working with Text Link Rules](#)

- [Creating and Editing Rules](#)
 - [Checking for Errors, Saving, and Cancelling](#)
 - [Processing Order for Rules](#)
-

Checking for Errors, Saving, and Cancelling

Applying Rule Changes

If you click outside of the rule editor or if you click Apply, the rule is automatically scanned for errors. If an error is found, you will need to fix it before moving on to another part of the application.

However, if less serious errors are detected, only a warning is given. For example, if your rule contains incomplete or unreferenced definitions to types or macros, a warning message is displayed. Once you click Apply, any uncorrected warnings cause a warning icon to appear to the left of the rule name in the tree in the left pane.

Applying a rule does not mean that your rule is permanently saved. Applying will cause the validation process to check for errors and warnings.

Saving Resources inside an Interactive Workbench Session

1. To save the changes you made to your resources during an interactive workbench session so you can get them next time you run your stream, you must:
 - Update your modeling node to make sure that you can get these same resources next time you execute your stream. See the topic [Updating Modeling Nodes and Saving](#) for more information. Then save your stream. To save your stream, do so in the main IBM® SPSS® Modeler window after updating the modeling node.
2. To save the changes you made to your resources during an interactive workbench session so that you can use them in other streams, you can:
 - Update the template you used or make a new one. See the topic [Making and Updating Templates](#) for more information. This will not save the changes for the current node (see previous step)
 - Or, update the TAP you used. See the topic [Updating Text Analysis Packages](#) for more information.

Saving Resources inside the Template Editor

1. First, publish the library. See the topic [Publishing Libraries](#) for more information.
2. Then, save the template through File > Save Resource Template in the menus.

Cancelling Rule Changes

1. If you wish to discard the changes, click Cancel in the editor pane.

Related information

- [Creating and Editing Macros](#)
 - [Working with Text Link Rules](#)
 - [Creating and Editing Rules](#)
 - [Disabling and Deleting Rules](#)
 - [Processing Order for Rules](#)
-

Processing Order for Rules

When text link analysis is performed during extraction, a "sentence" (clause, word, phrase) will be matched against each rule in turn until a match is found or all rules have been exhausted. Position in the tree dictates the order in which rules are tried. Best practice states that you should order your rules from most specific to most generic. The most specific ones should be at the top of the tree. To change the order of a specific rule or rule set, select Move up or Move down from the Rules and Macro Tree context menu or the up and down arrows in the toolbar.

If you are *in the source view*, you cannot change the order of the rules by moving them around in the editor. The higher up the rule appears in the source view, the sooner it is processed. We strongly recommend reordering rules only in the tree to avoid copy/paste issues.

Important! In previous versions of IBM® SPSS® Modeler Text Analytics, you were required to have a unique, numeric rule ID. Starting in version 18.4.0, you can only indicate processing order by moving a rule up or down in the tree, or by their position in the source view.

For example, suppose your text contains the following two sentences:

I love anchovies

I love anchovies and green peppers

In addition, suppose that two text link analysis rules exist with the following values:

Figure 1. 2 Example Rules

A	Element	Quantity	Example Token
1	Positive	Exactly 1	
2	mDet	0 or 1	
3	mTopic	Exactly 1	
4			
5			
6			
7			

B	Element	Quantity	Example Token
1	Positive	Exactly 1	
2	mDet	0 or 1	
3	mTopic	Exactly 1	
4	(SEP and or)	1 or 2	
5	mDet	0 or 1	
6	mTopic	Exactly 1	
7			

In the source view, the rule values might look like the following:

A: value = \$Positive \$mDet? \$mTopic

B: value = \$Positive \$mDet? \$mTopic (\$SEP|and|or){1,2} \$mDet? \$mTopic

If rule **A** is higher up in the tree (closer to the top) than rule **B**, then rule **A** will be processed first and the sentence *I love anchovies and green peppers* will be first matched by **\$Positive \$mDet? \$mTopic**, and it will produce an incomplete pattern output (**anchovies + like**) since it was matched by a rule that wasn't looking for 2 **\$mTopic** matches.

Therefore, to capture the true essence of the text, the most specific rule, in this case **B** must be placed higher in the tree than the more generic one, in this case rule **A**.

Related information

- [Creating and Editing Macros](#)
 - [Working with Text Link Rules](#)
 - [Creating and Editing Rules](#)
 - [Disabling and Deleting Rules](#)
 - [Checking for Errors, Saving, and Cancelling](#)
 - [Working with Macros](#)
 - [Disabling and Deleting Macros](#)
 - [Checking for Errors, Saving, and Cancelling](#)
 - [Special Macros: mTopic, mNonLingEntities, SEP](#)
-

Working with Rule Sets (Multiple Pass)

A rule set is a helpful way of grouping a related set of rules together in the Rules and Macro Tree so as to perform multiple pass processing. A rule set has no definition itself other than a name, and is used to organize your rules into meaningful groups. In some contexts, the text is too rich and varied to be processed in a single pass. For example, when working with security intelligence data, the text may contain links between individuals that are uncovered through contact methods (*x called y*), through family relationships (*y's brother-in-law x*), through exchange of money (*x wired \$100 to y*), and so on. In this case, it is helpful to create specialized sets of text link analysis rules, each of which is focused on a certain kind of relationship such as one for uncovering contacts, another for uncovering family members, and so on.

To create a rule set, select "Create Rule Set" from the Rules and Macro Tree context menu, or from the toolbar. You can then create new rules directly under a Rule Set node on the tree, or move existing rules to a Rule Set.

When you run an extraction using resources in which the rules are grouped into rule sets, the extraction engine is forced to make multiple passes through the text in order to match different kinds of patterns in each pass. In this way, a "sentence" can be matched to a rule in each rule set, whereas without a rule set it can only matched to a single rule.

Note: You can add up to 512 rules per rule set.

Creating New Rule Sets

1. From the menus, choose Tools > New Rule Set. Alternatively, click the New Rule Set icon in the tree toolbar. A rule set appears in the rule tree.
2. Add new rules to this rule set or move existing rules into the set.

Disabling Rule Sets

1. Right-click the rule set name in the tree.
2. From the context menus, choose Disable. The rule set icon becomes gray and all of the rules contained within that rule set are also disabled and ignored during processing.

Deleting Rule Sets

1. Right-click the rule set name in the tree.
2. From the context menus, choose Delete. The rule set and all the rules it contains are deleted from the resources.

Related information

- [About Text Link Rules](#)
 - [Where to work on text link rules](#)
 - [Where to Begin](#)
 - [When to Edit or Create Rules](#)
 - [Simulating Text Link Analysis Results](#)
 - [Navigating Rules and Macros in the Tree](#)
 - [Working with Macros](#)
 - [Creating and Editing Macros](#)
 - [Disabling and Deleting Macros](#)
 - [Checking for Errors, Saving, and Cancelling](#)
 - [Special Macros: mTopic, mNonLingEntities, SEP](#)
 - [Working with Text Link Rules](#)
 - [Supported Elements for Rules and Macros](#)
 - [Viewing and working in source mode](#)
-

Supported Elements for Rules and Macros

The following arguments are accepted for the value parameters in text link analysis rules and macros:

Macros

You can use a macro directly in a text link analysis rule or within another macro. If you are entering the macro name by hand or from within the source view (as opposed to selecting the macro name from a context menu), make sure to prefix the name with a dollar sign character (\$), such as \$mTopic. The macro name is case sensitive. You can choose from any macro defined in the current Text Link Rules tab when selecting macros through the context menus.

Types

You can use a type directly in a text link analysis rule or macro. If you are entering the type name by hand or in the source view (as opposed to selecting the type from a context menu), make sure to prefix the type name with a dollar sign character (\$), such as \$Person. The type name is case sensitive. If you use the context menus, you can choose from any type from the current set of resources being used.

If you reference an unrecognized type, you will receive a warning message, and the rule will have a warning icon in the Rules and Macro Tree until you correct it.

Literal Strings

To include information that was never extracted, you can define a literal string for which the extraction engine will search. All extracted words or phrases have been assigned to a type and for this reason, they cannot be used in literal strings. If you use a word that was extracted, it will be ignored, even if its type is <Unknown>.

A literal string can be one or more words. The following rules apply when defining a list of literal strings:

- Enclose the list of strings in parentheses such as (his). If there is a choice of literal strings then each string must be separated by the OR operator, such as (a|an|the) or (his|hers|its).
- Use single or compound words.
- Separate each word in the list by the | character, which is like a Boolean OR.
- Enter both singular and plural forms if you want to match both. Inflection is not automatically generated.
- Use lower case only.
- To reuse literal strings, define them as a macro and then use that macro in your other macros and text link analysis rules.
- If a string contains periods (full stops) or hyphens, you must include them. For example, to match a.k.a in the text, enter the periods along with the letters a.k.a as the literal string.

Exclusion Operator

Use ! as an exclusion operator to stop any expression of the negation from occupying a particular slot. You can only add an exclusion operator by hand through inline cell editing (double-click the cell in the Rule Value table or Macro Value table) or in the source view. For example, if you add `$mTopic @{0,2} !($Positive) $Budget` to your text link analysis rule, you are looking for text that contains (1) a term assigned to any of the types in the `mTopic` macro, (2) a word gap of zero to two words long, (3) no instances of a term assigned to the `<Positive>` type, and (4) a term assigned to the `<Budget>` type. This might capture "cars have an inflated price tag" but would ignore "store offers amazing discounts".

To use this operator, you must enter the exclamation point and parenthesis manually into the element cell by double-clicking the cell.

Word Gaps (<Any Token>)

A word gap, also referred to as `<Any Token>`, defines a numeric range of tokens that may be present between two elements. Word gaps are very useful when matching very similar phrases that may differ only slightly due to the presence of additional determiners, prepositional phrases, adjectives, or other such words.

Table 1. Example of the elements in a Rule Value table without a word gap

#	Element
1	Unknown
2	mBeHave
3	Positive

Note: In the source view this value is defined as: `$Unknown $mBeHave $Positive`

This value will match sentences like "*the hotel staff was nice*", where *hotel staff* belongs to type `<Unknown>`, *was* is under the macro `mBeHave` and *nice* is `<Positive>`. But it will not match "*the hotel staff was very nice*".

Table 2. Example of the elements in a Rule Value table with a `<Any Token>` word gap

#	Element
1	Unknown
2	mBeHave
3	
4	Positive

Note: In the source view this value is defined as: `$Unknown $mBeHave @{0,1} $Positive`

If you add a word gap to your rule value, it will match both "*the hotel staff was nice*" and "*the hotel staff was very nice*".

In the source view or with inline editing, the syntax for a word gap is `@{#, #}`, where @ signifies a word gap and the `{#, #}` defines the minimum and maximum of words accepted between the preceding element and following element. For example, `@{1,3}` means that a match can be made between the two defined elements if there is at least one word present but no more than three words appearing between those two elements. `@{0,3}` means that a match can be made between the two defined elements if there is 0, 1, 2 or 3 words present but no more than three words.

Related information

- [About Text Link Rules](#)
- [Where to work on text link rules](#)
- [Where to Begin](#)
- [When to Edit or Create Rules](#)
- [Simulating Text Link Analysis Results](#)
- [Navigating Rules and Macros in the Tree](#)
- [Working with Macros](#)
- [Creating and Editing Macros](#)
- [Disabling and Deleting Macros](#)
- [Checking for Errors, Saving, and Cancelling](#)
- [Special Macros: mTopic, mNonLingEntities, SEP](#)
- [Working with Text Link Rules](#)
- [Working with Rule Sets \(Multiple Pass\)](#)
- [Viewing and working in source mode](#)

Viewing and working in source mode

For each rule and macro the TLA editor generates the underlying source code that is used by the Extractor for matching and producing TLA output. If you prefer to work with the code itself, you can view this source code and edit it directly by clicking the “View Source” button at the top of the Editor. The Source view will jump to and highlight the currently selected rule or macro. However, we recommend using the editor panes to reduce the chance of errors.

When you have finished viewing or editing the source, click Exit Source. If you generate invalid syntax for a rule, you will be required to fix it before you exit the source view.

Important: If you edit in the source view, we strongly recommend that you edit rules and macros one at a time. After editing a macro, please validate the results by extracting. If you are satisfied with the result, we recommend that you save the template before making another change. If you are not satisfied with the result or an error occurs, revert to your saved resources.

Macros in the Source View

```
[macro]
name = macro_name
value = ([type_name|macro_name|literal_string|word_gap])
```

Table 1. Macro entries

[macro]	Each macro must begin with the line marked [macro] to denote the beginning of a macro.
name	The name of the macro definition. Each name must be unique.
value	A combination of one or more types, literal strings, word gaps, or macros. See the topic Supported Elements for Rules and Macros for more information. When combining arguments, you must use parentheses () to group the arguments and the character to indicate a Boolean OR.

In addition to the guidelines and syntax covered in the section on Macros, the source view has a few additional guidelines that aren't required when working in the editor view. Macros must also respect the following when working in source mode:

- Each macro must begin with the line marked [macro] to denote the beginning of a macro.
- To disable an element, place a comment indicator (#) before each line.

Example. This example defines a macro called mTopic. The value for mTopic is the presence of a term matching one of the following types: <Product>, <Person>, <Location>, <Organization>, <Budget>, or <Unknown>.

```
[macro]
name=mTopic
value=($Unknown|$Product|$Person|$Location|$Organization|$Budget|$Currency)
```

Rules in the Source View

```
[pattern(ID)]
name = pattern_name
value = [$type_name|macro_name|word_gaps|literal_strings]
output = $digit[\t]#${digit[\t]}$digit[\t]#${digit[\t]}$digit[\t]
```

Table 2. Rule entries

[pattern(<ID>)]	Indicates the start of a text link analysis rule and provides a unique numerical ID use to determine processing order.
name	Provides a unique name for this text link analysis rule.
value	Provides the syntax and arguments to be matched to the text. See the topic Supported Elements for Rules and Macros for more information.
output	The output format for the resulting matched patterns discovered in the text. The output does not always resemble the exact original position of elements in the source text. Additionally, it is possible to have multiple output lines for a given text link analysis rule by placing each output on a separate line. Syntax for output: <ul style="list-style-type: none">• Separate output with the tab code \t, such as \$1\t#1\t\$3\t#3• \$ and a number calls for the term found matching the argument defined in the value parameter in that position. So \$1 means the term matching the first argument defined for the value.• # and a number calls for the type name of the element in that position. If an item is a list of literal strings, the type <Unknown> will be assigned.• A value of Null\tNull will not create any output.

In addition to the guidelines and syntax covered in the section on Rules, the source view has a few additional guidelines that aren't required when working in the editor view. Rules must also respect the following when working in source mode:

- Whenever two or more elements are defined, they must be enclosed in parentheses whether or not they are optional (for example, `($Negative|$Positive)` or `($mCoord|$SEP)?`). `$SEP` represents a comma.
- The first element in a text link analysis rule cannot be an optional element. For example, you cannot begin with `value = $mTopic?` or `value = @{0,1}.`
- It is possible to associate a quantity (or instance count) to a token. This is useful in writing only one rule that encompasses all cases instead of writing a separate rule for each case. For example, you may use the literal string `($SEP|and)` if you are trying to match either , (comma) or `and`. If you extend this by adding a quantity so that the literal string becomes `($SEP|and){1,2}`, you will now match any of the following instances: " , " `and` " , `and`".
- Spaces are not supported between the macro name and the `$` and `?` characters in the text link analysis rule `value`.
- Spaces are not supported in the text link analysis rule `output`.
- To disable an element, place a comment indicator (#) before each line.

Example. Let's suppose your resources contain the following TLA text link analysis rule and that you have enabled the extraction of TLA results:

```
## Jean Doe was the former HR director of IBM in France
[pattern(201)]
name= 1_201
value = $Person ($SEP|$mDet|$mSupport|as|then){1,2} @{0,1} $Function
        (of|with|for|in|to|at) @{0,1} $Organization @{0,2} $Location
output = $1\t#1\t$4\t#4\t$7\t#7\t$9\t#9
```

Whenever you extract, the extraction engine will read each sentence and will try to match the following sequence:

Table 3. Extraction sequence example

Position	Description of the arguments
1	The name of a person (<code>\$Person</code>),
2	One or two of the following: comma (<code>\$SEP</code>), determiner (<code>\$mDet</code>), auxiliary verb (<code>\$mSupport</code>), the strings “ <code>then</code> ” or “ <code>as</code> ”,
3	0 or 1 word (@{0,1})
4	A function (<code>\$Function</code>)
5	One of the following strings: “ <code>of</code> ”, “ <code>with</code> ”, “ <code>for</code> ”, “ <code>in</code> ”, “ <code>to</code> ”, or “ <code>at</code> ”,
6	0 or 1 word (@{0,1})
7	The name of an organization (<code>\$Organization</code>)
8	0, 1, or 2 words (@{0,2})
9	The name of a location (<code>\$Location</code>)

This sample text link analysis rule would match sentences or phrases like:

Jean Doe, the HR director of IBM in France

Jean Doe was the former HR director of IBM in France

IBM appointed Jean Doe as the HR director of IBM in France

This sample text link analysis rule would produce the following output:

```
jean doe <Person> hr director <Function> ibm <Organization> france <Location>
```

Where:

- `jean doe` is the term corresponding to `$1` (the first element in the text link analysis rule) and `<Person>` is the type for `jean doe` (#1),
- `hr director` is the term corresponding to `$4` (the 4th element in the text link analysis rule) and `<Function>` is the type for `hr director` (#4),
- `ibm` is the term corresponding to `$7` (the 7th element in the text link analysis rule) and `<Organization>` is the type for `ibm`. (#7),
- `france` is the term corresponding to `$9` (the 9th element in the text link analysis rule) and `<Location>` is the type for `france` (#9)

Rule Sets in the Source View

```
[set(<ID>)]
```

Where `[set (<ID>)]` indicates the start of a rule set and provides a unique numerical ID use to determine processing order of the sets.

Example. The following sentence contains information about individuals, their function within a company, and also the merge/acquisition activities of that company.

```
Org1 Inc has entered into a definitive merger agreement with Org2 Ltd, said John Doe, CEO of Org2 Ltd.
```

You could write one rule with several outputs to handle all possible output such as:

```
## Org1 Inc entered into a definitive merger agreement with Org2 Ltd, said
John Doe, CEO of Org2 Ltd.
```

```
[pattern(020)]
name=020
value = $Organization @{0,4} $ActionNouns @{0,6} $mOrg @{1,2}
$Person @{0,2} $Function @{0,1} $Organization
output = $1\t#1\t$3\t#3\t$5\t#5
output = $7\t#7\t$9\t#9\t$11\t#11
```

which would produce the following 2 output patterns:

- org1 inc<Organization> + merges with <ActiveVerb> + org2
ltd<Organization>
- john doe <Person> + ceo <Function> + org2 ltd<Organization>

Important! Keep in mind that other linguistic handling operations are performed during the extraction of TLA patterns. In this case, `merges` is grouped under `merges`

`with` during the synonym grouping phase of the extraction process. And since `merges` belongs to `<ActiveVerb>` type, this type name is what appears in the final TLA pattern output. So when the output reads `t$3\t#3`, this means that the pattern will ultimately display the final concept for the third element and the final type for the third element after all linguistic processing is applied (synonyms and other groupings).

Instead of writing complex rules like the preceding, it can be easier to manage and work with two rules. The first is specialized in finding out mergers/acquisitions between companies:

```
[set(1)]
## Org1 Inc has entered into a definitive merger agreement with Org2 Ltd
[pattern(44)]
name=firm + action + firm_0044
value=$mOrg @{0,20} $ActionNouns @{0,6} $mOrg
output(1)=$1\t#1\t$3\t#3\t$5\t#5
```

which would produce `org1 inc<Organization> + merges with <ActiveVerb> + org2 ltd <Organization>`

The second is specialized in individual/function/company:

```
[set(2)]
## said John Doe, CEO of Org2 Ltd
[pattern(52)]
name=individual + role + firm_0007
value=$Person @{0,3} $mFunction (at|of)? ($mOrg|$Media|$Unknown)
output(1)=$1\t#1\t$3\tFunction\t$5\t#5
```

which would produce `john doe <Person> + ceo <Function> + org2 ltd <Organization>`

About IBM SPSS Modeler

IBM® SPSS® Modeler is a set of data mining tools that enable you to quickly develop predictive models using business expertise and deploy them into business operations to improve decision making. Designed around the industry-standard CRISP-DM model, IBM SPSS Modeler supports the entire data mining process, from data to better business results.

IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics. The methods available on the Modeling palette allow you to derive new information from your data and to develop predictive models. Each method has certain strengths and is best suited for particular types of problems.

SPSS Modeler can be purchased as a standalone product, or used as a client in combination with SPSS Modeler Server. A number of additional options are also available, as summarized in the following sections. For more information, see <https://www.ibm.com/analytics/us/en/technology/spss/>.

- [IBM SPSS Modeler Products](#)
- [IBM SPSS Modeler Editions](#)
- [Documentation](#)
- [Application examples](#)
- [Demos Folder](#)
- [License tracking](#)

IBM SPSS Modeler Products

The IBM® SPSS® Modeler family of products and associated software comprises the following.

- IBM SPSS Modeler
 - IBM SPSS Modeler Server
 - IBM SPSS Modeler Administration Console (included with IBM SPSS Deployment Manager)
 - IBM SPSS Modeler Batch
 - IBM SPSS Modeler Solution Publisher
 - IBM SPSS Modeler Server adapters for IBM SPSS Collaboration and Deployment Services
- [IBM SPSS Modeler](#)**
[IBM SPSS Modeler Server](#)
[IBM SPSS Modeler Administration Console](#)
[IBM SPSS Modeler Batch](#)
[IBM SPSS Modeler Solution Publisher](#)
[IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services](#)
-

IBM SPSS Modeler

SPSS® Modeler is a functionally complete version of the product that you install and run on your personal computer. You can run SPSS Modeler in local mode as a standalone product, or use it in distributed mode along with IBM® SPSS Modeler Server for improved performance on large data sets.

With SPSS Modeler, you can build accurate predictive models quickly and intuitively, without programming. Using the unique visual interface, you can easily visualize the data mining process. With the support of the advanced analytics embedded in the product, you can discover previously hidden patterns and trends in your data. You can model outcomes and understand the factors that influence them, enabling you to take advantage of business opportunities and mitigate risks.

SPSS Modeler is available in two editions: SPSS Modeler Professional and SPSS Modeler Premium. See the topic [IBM SPSS Modeler Editions](#) for more information.

IBM SPSS Modeler Server

SPSS® Modeler uses a client/server architecture to distribute requests for resource-intensive operations to powerful server software, resulting in faster performance on larger data sets.

SPSS Modeler Server is a separately-licensed product that runs continually in distributed analysis mode on a server host in conjunction with one or more IBM® SPSS Modeler installations. In this way, SPSS Modeler Server provides superior performance on large data sets because memory-intensive operations can be done on the server without downloading data to the client computer. IBM SPSS Modeler Server also provides support for SQL optimization and in-database modeling capabilities, delivering further benefits in performance and automation.

IBM SPSS Modeler Administration Console

The Modeler Administration Console is a graphical user interface for managing many of the SPSS® Modeler Server configuration options, which are also configurable by means of an options file. The console is included in IBM® SPSS Deployment Manager, can be used to monitor and configure your SPSS Modeler Server installations, and is available free-of-charge to current SPSS Modeler Server customers. The application can be installed only on Windows computers; however, it can administer a server installed on any supported platform.

IBM® SPSS Modeler Batch

While data mining is usually an interactive process, it is also possible to run SPSS® Modeler from a command line, without the need for the graphical user interface. For example, you might have long-running or repetitive tasks that you want to perform with no user intervention. SPSS Modeler Batch is a special version of the product that provides support for the complete analytical capabilities of SPSS Modeler without access to the regular user interface. SPSS Modeler Server is required to use SPSS Modeler Batch.

IBM SPSS Modeler Solution Publisher

SPSS® Modeler Solution Publisher is a tool that enables you to create a packaged version of an SPSS Modeler stream that can be run by an external runtime engine or embedded in an external application. In this way, you can publish and deploy complete SPSS Modeler streams for use

in environments that do not have SPSS Modeler installed. SPSS Modeler Solution Publisher is distributed as part of the IBM® SPSS Collaboration and Deployment Services - Scoring service, for which a separate license is required. With this license, you receive SPSS Modeler Solution Publisher Runtime, which enables you to execute the published streams.

For more information about SPSS Modeler Solution Publisher, see the IBM SPSS Collaboration and Deployment Services documentation. The IBM SPSS Collaboration and Deployment Services IBM Documentation contains sections called "IBM SPSS Modeler Solution Publisher" and "IBM SPSS Analytics Toolkit."

IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services

A number of adapters for IBM® SPSS® Collaboration and Deployment Services are available that enable SPSS Modeler and SPSS Modeler Server to interact with an IBM SPSS Collaboration and Deployment Services repository. In this way, an SPSS Modeler stream deployed to the repository can be shared by multiple users, or accessed from the thin-client application IBM SPSS Modeler Advantage. You install the adapter on the system that hosts the repository.

IBM SPSS Modeler Editions

SPSS® Modeler is available in the following editions.

SPSS Modeler Professional

SPSS Modeler Professional provides all the tools you need to work with most types of structured data, such as behaviors and interactions tracked in CRM systems, demographics, purchasing behavior and sales data.

SPSS Modeler Premium

SPSS Modeler Premium is a separately-licensed product that extends SPSS Modeler Professional to work with specialized data and with unstructured text data. SPSS Modeler Premium includes IBM® SPSS Modeler Text Analytics:

IBM SPSS Modeler Text Analytics uses advanced linguistic technologies and Natural Language Processing (NLP) to rapidly process a large variety of unstructured text data, extract and organize the key concepts, and group these concepts into categories. Extracted concepts and categories can be combined with existing structured data, such as demographics, and applied to modeling using the full suite of IBM SPSS Modeler data mining tools to yield better and more focused decisions.

IBM SPSS Modeler Subscription

IBM SPSS Modeler Subscription provides all the same predictive analytics capabilities as the traditional IBM SPSS Modeler client. With the Subscription edition, you can download product updates regularly.

Documentation

Documentation is available from the Help menu in SPSS® Modeler. This opens the online IBM Documentation, which is always available outside the product.

Complete documentation for each product (including installation instructions) is also available in PDF format at <https://www.ibm.com/support/pages/spss-modeler-184-documentation>.

- [SPSS Modeler Professional documentation](#)
- [SPSS Modeler Premium Documentation](#)

SPSS Modeler Professional documentation

The SPSS® Modeler Professional documentation suite (excluding installation instructions) is as follows.

- **IBM® SPSS Modeler User's Guide.** General introduction to using SPSS Modeler, including how to build data streams, handle missing values, build CLEM expressions, work with projects and reports, and package streams for deployment to IBM SPSS Collaboration and Deployment Services or IBM SPSS Modeler Advantage.
- **IBM SPSS Modeler Source, Process, and Output Nodes.** Descriptions of all the nodes used to read, process, and output data in different formats. Effectively this means all nodes other than modeling nodes.
- **IBM SPSS Modeler Modeling Nodes.** Descriptions of all the nodes used to create data mining models. IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics.
- **IBM SPSS Modeler Algorithms Guide.** Descriptions of the mathematical foundations of the modeling methods used in IBM SPSS Modeler. This guide is available in PDF format only.
- **IBM SPSS Modeler Applications Guide.** The examples in this guide provide brief, targeted introductions to specific modeling methods and techniques. An online version of this guide is also available from the Help menu. See the topic [Application examples](#) for more information.
- **IBM SPSS Modeler Python Scripting and Automation.** Information on automating the system through Python scripting, including the properties that can be used to manipulate nodes and streams.
- **IBM SPSS Modeler Deployment Guide.** Information on running IBM SPSS Modeler streams as steps in processing jobs under IBM SPSS Deployment Manager.
- **IBM SPSS Modeler In-Database Mining Guide.** Information on how to use the power of your database to improve performance and extend the range of analytical capabilities through third-party algorithms.
- **IBM SPSS Modeler Server Administration and Performance Guide.** Information on how to configure and administer IBM SPSS Modeler Server.
- **IBM SPSS Deployment Manager User Guide.** Information on using the administration console user interface included in the Deployment Manager application for monitoring and configuring IBM SPSS Modeler Server.
- **IBM SPSS Modeler CRISP-DM Guide.** Step-by-step guide to using the CRISP-DM methodology for data mining with SPSS Modeler.
- **IBM SPSS Modeler Batch User's Guide.** Complete guide to using IBM SPSS Modeler in batch mode, including details of batch mode execution and command-line arguments.

SPSS Modeler Premium Documentation

The SPSS® Modeler Premium documentation suite (excluding installation instructions) is as follows.

- **SPSS Modeler Text Analytics User's Guide.** Information on using text analytics with SPSS Modeler, covering the text mining nodes, interactive workbench, templates, and other resources.

Application examples

While the data mining tools in SPSS® Modeler can help solve a wide variety of business and organizational problems, the application examples provide brief, targeted introductions to specific modeling methods and techniques. The data sets used here are much smaller than the enormous data stores managed by some data miners, but the concepts and methods that are involved are scalable to real-world applications.

To access the examples, click Application Examples on the Help menu in SPSS Modeler.

The data files and sample streams are installed in the Demos folder under the product installation directory. For more information, see [Demos Folder](#).

The following examples are available:

- [Introduction to Modeling](#)
- [Automated Modeling for a Flag Target](#)
- [Automated Modeling for a Continuous Target](#)
- [Automated Data Preparation \(ADP\)](#)
- [Preparing Data for Analysis \(Data Audit\)](#)
- [Drug Treatments \(Exploratory Graphs/C5.0\)](#)
- [Screening Predictors \(Feature Selection\)](#)
- [Reducing Input Data String Length \(Reclassify Node\)](#)
- [Modeling Customer Response \(Decision List\)](#)
- [Classifying Telecommunications Customers \(Multinomial Logistic Regression\)](#)
- [Telecommunications Churn \(Binomial Logistic Regression\)](#)
- [Forecasting Bandwidth Utilization \(Time Series\)](#)
- [Forecasting Catalog Sales \(Time Series\)](#)
- [Making Offers to Customers \(Self-Learning\)](#)
- [Predicting Loan Defaulters \(Bayesian Network\)](#)
- [Retraining a Model on a Monthly Basis \(Bayesian Network\)](#)
- [Retail Sales Promotion \(Neural Net/C&RT\)](#)
- [Condition Monitoring \(Neural Net/C5.0\)](#)

- [Classifying Telecommunications Customers \(Discriminant Analysis\)](#)
- [Analyzing interval-censored survival data \(Generalized Linear Models\)](#)
- [Using Poisson regression to analyze ship damage rates \(Generalized Linear Models\)](#)
- [Fitting a Gamma regression to car insurance claims \(Generalized Linear Models\)](#)
- [Classifying Cell Samples \(SVM\)](#)
- [Tracking the Expected Number of Customers Retained](#)
- [Market Basket Analysis \(Rule Induction/C5.0\)](#)
- [Assessing New Vehicle Offerings \(KNN\)](#)
- [Uncovering causal relationships in business metrics \(TCM\)](#)

After you open an example, click the Next button in the lower left corner of the tutorial page to move forward through the example.

A PDF version of the Applications Guide is also available. For more information, see [Documentation](#).

Demos Folder

The data files and sample streams that are used with the application examples are installed in the Demos folder under the product installation directory (for example: C:\Program Files\IBM\SPSS\Modeler\<version>\Demos). This folder can also be accessed from the IBM SPSS® Modeler program group on the Windows Start menu, or by clicking Demos on the list of recent directories in the File->Open Stream dialog box.

License tracking

When you use SPSS® Modeler, license usage is tracked and logged at regular intervals. The license metrics that are logged are AUTHORIZED_USER and CONCURRENT_USER, and the type of metric that is logged depends on the type of license that you have for SPSS Modeler.

The log files that are produced can be processed by the IBM License Metric Tool, from which you can generate license usage reports.

The license log files are created in the same directory where SPSS Modeler Client log files are recorded (by default, %ALLUSERSPROFILE%\IBM\SPSS\Modeler\<version>/log).

Product overview

- [Getting started](#)
- [Starting IBM SPSS Modeler](#)
- [IBM SPSS Modeler Interface at a Glance](#)

Getting started

As a data mining application, IBM® SPSS® Modeler offers a strategic approach to finding useful relationships in large data sets. In contrast to more traditional statistical methods, you do not necessarily need to know what you are looking for when you start. You can explore your data, fitting different models and investigating different relationships, until you find useful information.

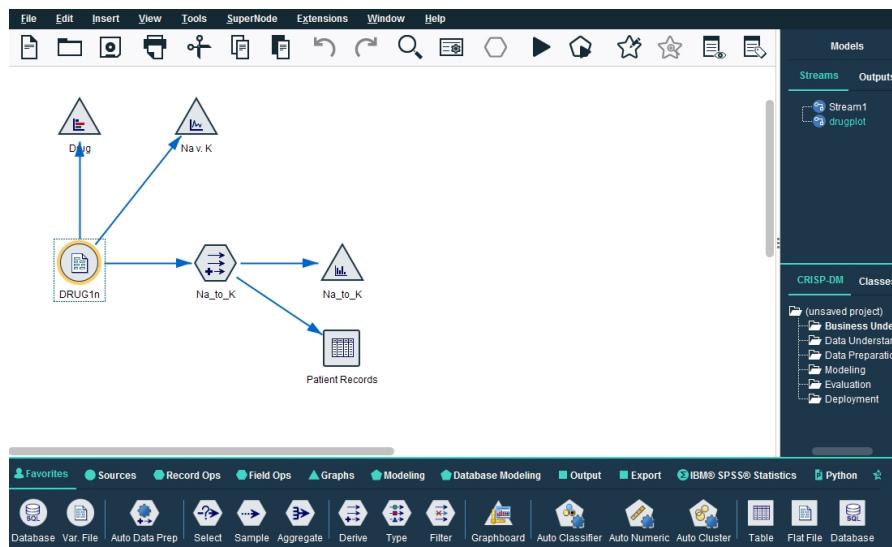
Starting IBM® SPSS® Modeler

To start the application, click:

Start->[All] Programs->IBM SPSS Modeler <version>->IBM SPSS Modeler <version>

The main window is displayed after a few seconds.

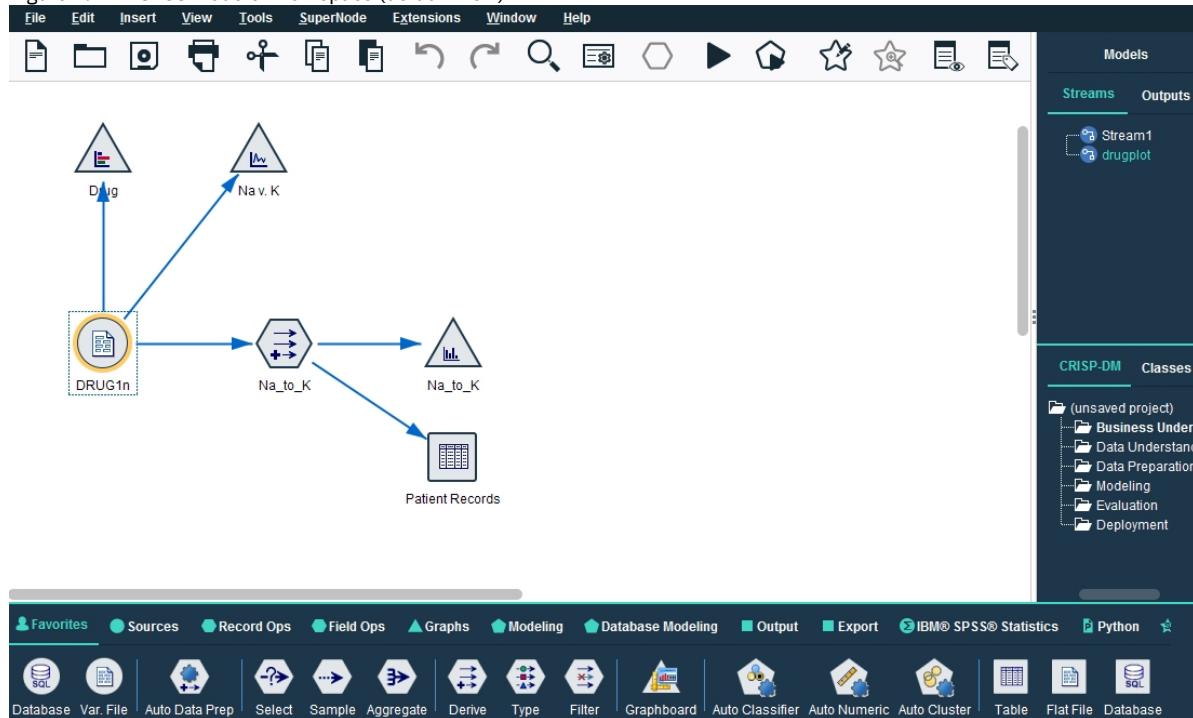
Figure 1. IBM SPSS Modeler main application window



IBM SPSS Modeler Interface at a Glance

At each point in the data mining process, the easy-to-use IBM® SPSS® Modeler interface invites your specific business expertise. Modeling algorithms, such as prediction, classification, segmentation, and association detection, ensure powerful and accurate models. Model results can easily be deployed and read into databases, IBM SPSS Statistics, and a wide variety of other applications.

Figure 1. IBM SPSS Modeler workspace (default view)



Working with IBM SPSS Modeler is a three-step process of working with data.

- First, you read data into IBM SPSS Modeler.
- Next, you run the data through a series of manipulations.
- Finally, you send the data to a destination.

This sequence of operations is known as a **data stream** because the data flows record by record from the source through each manipulation and, finally, to the destination--either a model or type of data output.

Figure 2. A simple stream

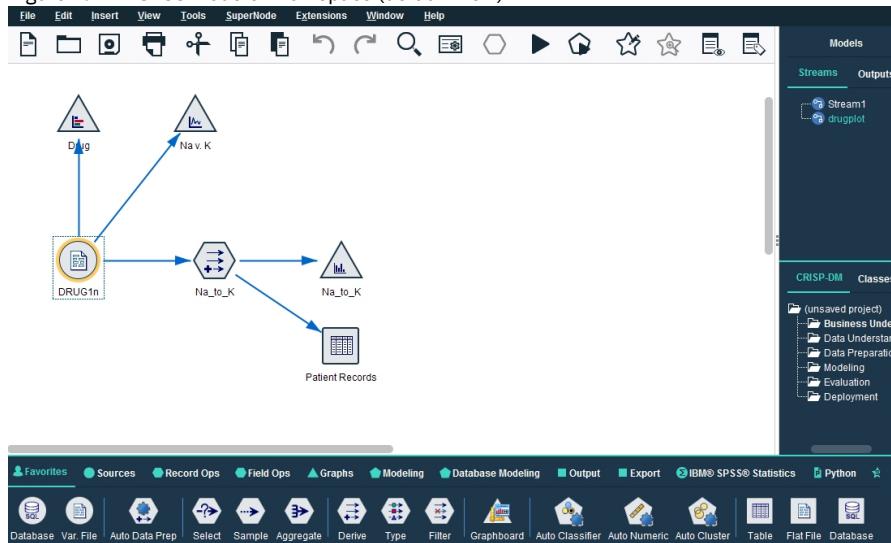


- [IBM SPSS Modeler Stream Canvas](#)
- [Nodes palette](#)
- [IBM SPSS Modeler Managers](#)
- [IBM SPSS Modeler Projects](#)
- [Using the Mouse in IBM SPSS Modeler](#)

IBM SPSS Modeler Stream Canvas

The stream canvas is the largest area of the IBM® SPSS® Modeler window and is where you will build and manipulate data streams.

Figure 1. IBM SPSS Modeler workspace (default view)



Streams are created by drawing diagrams of data operations relevant to your business on the main canvas in the interface. Each operation is represented by an icon or **node**, and the nodes are linked together in a **stream** representing the flow of data through each operation.

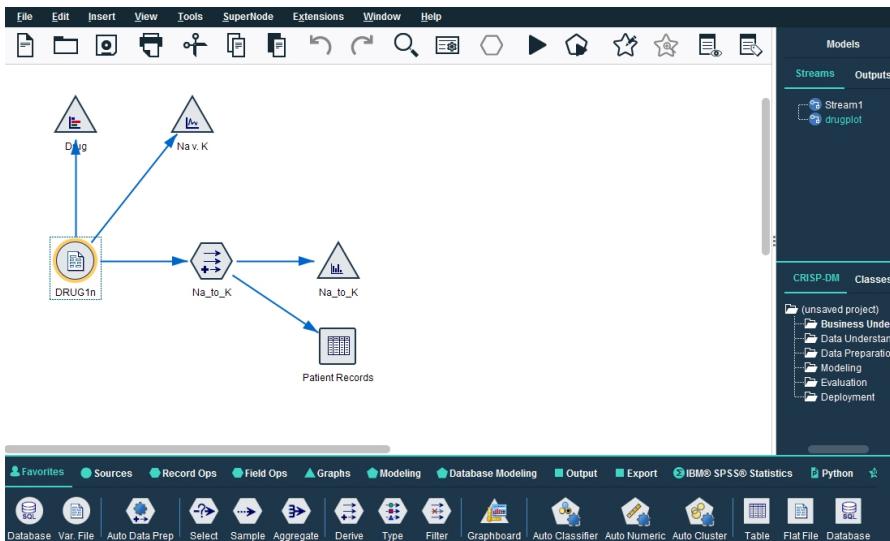
You can work with multiple streams at one time in IBM SPSS Modeler, either in the same stream canvas or by opening a new stream canvas. During a session, streams are stored in the Streams manager, at the upper right of the IBM SPSS Modeler window.

Note: If using a MacBook with the built-in trackpad's Force Click and haptic feedback setting enabled, dragging and dropping nodes from the nodes palette to the stream canvas can result in duplicate nodes being added to the canvas. To avoid this issue, we recommend disabling the Force Click and haptic feedback trackpad system preference.

Nodes palette

Most of the data and modeling tools in SPSS® Modeler are available from the *Nodes Palette*, across the bottom of the window below the stream canvas.

Figure 1. SPSS Modeler workspace (default view)



For example, the Record Ops palette tab contains nodes that you can use to perform operations on the data *records*, such as selecting, merging, and appending.

To add nodes to the canvas, double-click icons from the Nodes Palette or drag them onto the canvas. You then connect them to create a *stream*, representing the flow of data.

Each palette tab contains a collection of related nodes used for different phases of stream operations, such as:

- Sources nodes bring data into SPSS Modeler.
- Record Ops nodes perform operations on data *records*, such as selecting, merging, and appending.
- Field Ops nodes perform operations on data *fields*, such as filtering, deriving new fields, and determining the measurement level for given fields.
- Graphs nodes graphically display data before and after modeling. Graphs include plots, histograms, web nodes, and evaluation charts.
- Modeling nodes use the modeling algorithms available in SPSS Modeler, such as neural nets, decision trees, clustering algorithms, and data sequencing.
- Database Modeling nodes use the modeling algorithms available in Microsoft SQL Server, IBM Db2, and Oracle and Netezza databases.
- Output nodes produce various output for data, charts, and model results that can be viewed in SPSS Modeler.
- Export nodes produce various output that can be viewed in external applications, such as IBM® SPSS Data Collection or Excel.
- **IBM SPSS Statistics** nodes import data from, or export data to, IBM SPSS Statistics, as well as running IBM SPSS Statistics procedures.
- Python nodes can be used to run Python algorithms.
- Spark nodes can be used to run Spark algorithms.

As you become more familiar with SPSS Modeler, you can customize the palette contents for your own use.

On the left side of the Nodes Palette, you can filter the nodes that display by selecting Supervised, Association, or Segmentation.

Located below the Nodes Palette, a report pane provides feedback on the progress of various operations, such as when data is being read into the data stream. Also located below the Nodes Palette, a status pane provides information on what the application is currently doing, as well as indications of when user feedback is required.

Note: If using a MacBook with the built-in trackpad's Force Click and haptic feedback setting enabled, dragging and dropping nodes from the nodes palette to the stream canvas can result in duplicate nodes being added to the canvas. To avoid this issue, we recommend disabling the Force Click and haptic feedback trackpad system preference.

IBM SPSS Modeler Managers

At the top right of the window is the managers pane. This has three tabs, which are used to manage streams, output and models.

You can use the Streams tab to open, rename, save, and delete the streams created in a session.

Figure 1. Streams tab



Figure 2. Outputs tab



The Outputs tab contains a variety of files, such as graphs and tables, produced by stream operations in IBM® SPSS® Modeler. You can display, save, rename, and close the tables, graphs, and reports listed on this tab.

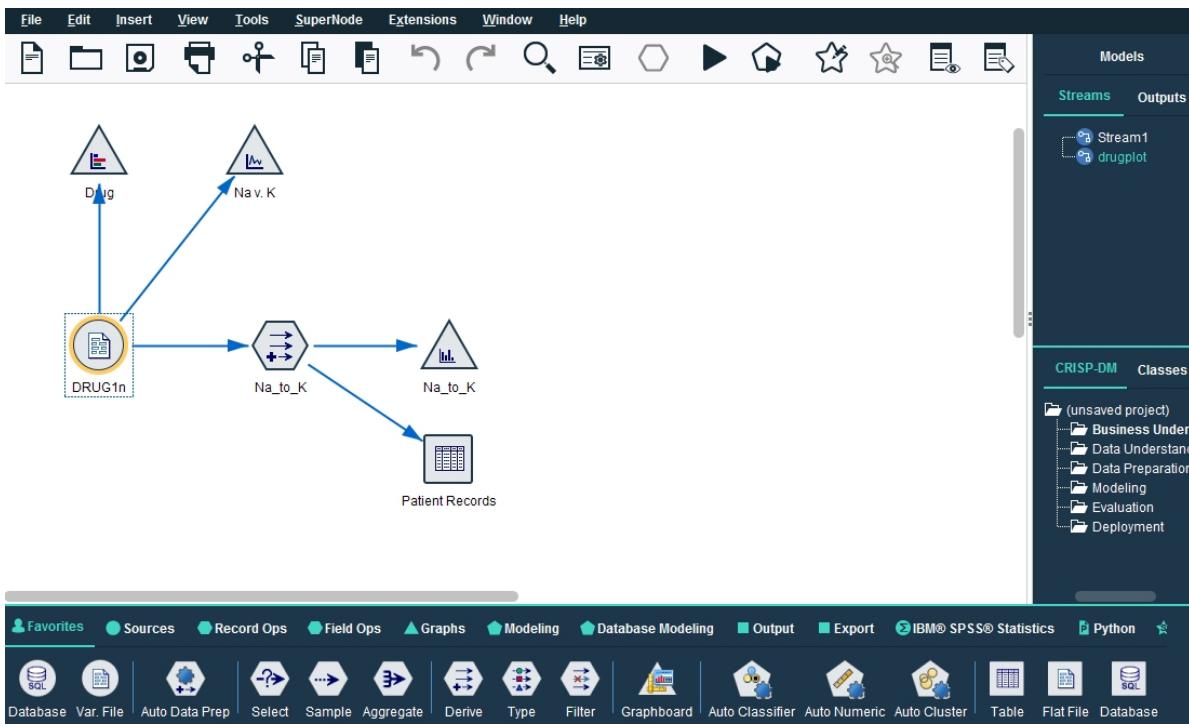
Figure 3. Models tab containing model nuggets



The Models tab is the most powerful of the manager tabs. This tab contains all model **nuggets**, which contain the models generated in IBM SPSS Modeler, for the current session. These models can be browsed directly from the Models tab or added to the stream in the canvas.

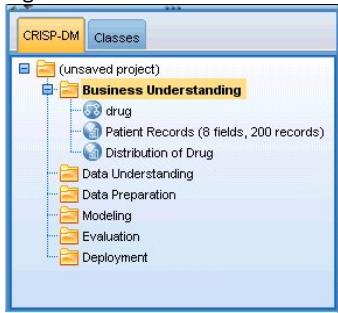
IBM SPSS Modeler Projects

Figure 1. IBM SPSS Modeler workspace (default view)



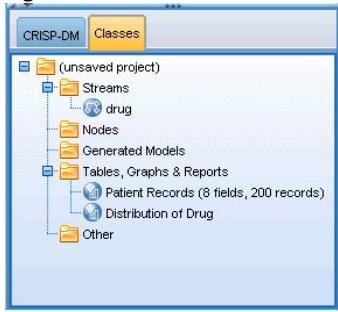
On the lower right side of the window is the project pane, used to create and manage data mining **projects** (groups of files related to a data mining task). There are two ways to view projects you create in IBM® SPSS® Modeler—in the **Classes** view and the **CRISP-DM** view.

Figure 2. CRISP-DM view



The CRISP-DM tab provides a way to organize projects according to the Cross-Industry Standard Process for Data Mining, an industry-proven, nonproprietary methodology. For both experienced and first-time data miners, using the CRISP-DM tool will help you to better organize and communicate your efforts.

Figure 3. Classes view



The Classes tab provides a way to organize your work in IBM SPSS Modeler categorically—by the types of objects you create. This view is useful when taking inventory of data, streams, and models.

Using the Mouse in IBM SPSS Modeler

The most common uses of the mouse in IBM® SPSS® Modeler include the following:

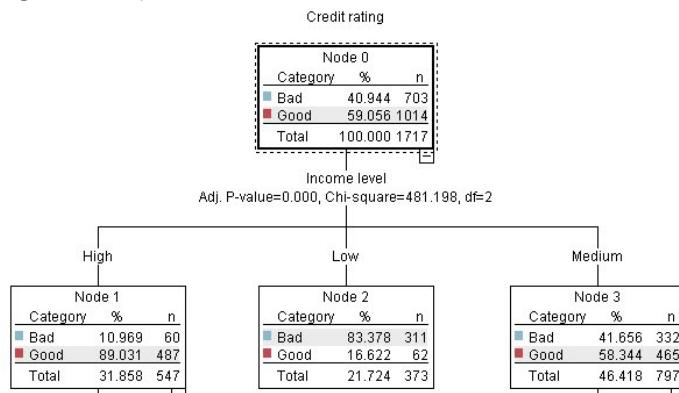
- **Single-click.** Use either the right or left mouse button to select options from menus, open pop-up menus, and access various other standard controls and options. Click and hold the button to move and drag nodes.
- **Double-click.** Double-click using the left mouse button to place nodes on the stream canvas and edit existing nodes.
- **Middle-click.** Click the middle mouse button and drag the cursor to connect nodes on the stream canvas. Double-click the middle mouse button to disconnect a node. If you do not have a three-button mouse, you can simulate this feature by pressing the Alt key while clicking and dragging the mouse.

Introduction to Modeling

A model is a set of rules, formulas, or equations that can be used to predict an outcome based on a set of input fields or variables. For example, a financial institution might use a model to predict whether loan applicants are likely to be good or bad risks, based on information that is already known about past applicants.

The ability to predict an outcome is the central goal of predictive analytics, and understanding the modeling process is the key to using IBM® SPSS® Modeler.

Figure 1. A simple decision tree model



This example uses a **decision tree** model, which classifies records (and predicts a response) using a series of decision rules, for example:

```
IF income = Medium
AND cards < 5
THEN -> 'Good'
```

While this example uses a CHAID (Chi-squared Automatic Interaction Detection) model, it is intended as a general introduction, and most of the concepts apply broadly to other modeling types in IBM SPSS Modeler.

To understand any model, you first need to understand the data that go into it. The data in this example contain information about the customers of a bank. The following fields are used:

Field name	Description
Credit_rating	Credit rating: 0=Bad, 1=Good, 9=missing values
Age	Age in years
Income	Income level: 1=Low, 2=Medium, 3=High
Credit_cards	Number of credit cards held: 1=Less than five, 2=Five or more
Education	Level of education: 1=High school, 2=College
Car_loans	Number of car loans taken out: 1=None or one, 2=More than two

The bank maintains a database of historical information on customers who have taken out loans with the bank, including whether or not they repaid the loans (Credit rating = Good) or defaulted (Credit rating = Bad). Using this existing data, the bank wants to build a model that will enable them to predict how likely future loan applicants are to default on the loan.

Using a decision tree model, you can analyze the characteristics of the two groups of customers and predict the likelihood of loan defaults.

This example uses the stream named *modelingintro.str*, available in the *Demos* folder under the *streams* subfolder. The data file is *tree_credit.sav*. See the topic [Demos Folder](#) for more information.

Let's take a look at the stream.

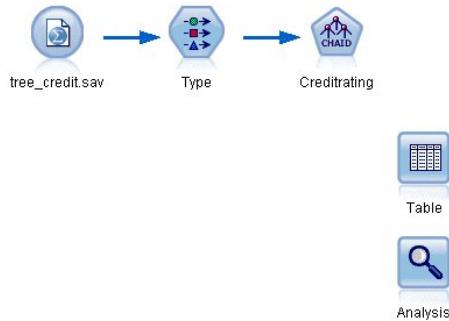
1. Choose the following from the main menu:
File->Open Stream
2. Click the gold nugget icon on the toolbar of the Open dialog box and choose the Demos folder.
3. Double-click the *streams* folder.
4. Double-click the file named *modelingintro.str*.

[Next](#)

- [Building the Stream](#)
- [Browsing the Model](#)
- [Evaluating the Model](#)
- [Scoring records](#)
- [Summary](#)

Building the Stream

Figure 1. Modeling stream



To build a stream that will create a model, we need at least three elements:

- A source node that reads in data from some external source, in this case an IBM® SPSS® Statistics data file.
- A source or Type node that specifies field properties, such as measurement level (the type of data that the field contains), and the role of each field as a target or input in modeling.
- A modeling node that generates a model nugget when the stream is run.

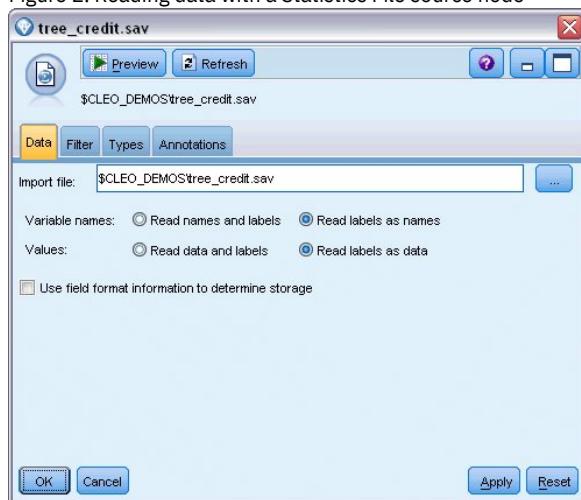
In this example, we're using a CHAID modeling node. CHAID, or Chi-squared Automatic Interaction Detection, is a classification method that builds decision trees by using a particular type of statistics known as chi-square statistics to work out the best places to make the splits in the decision tree.

If measurement levels are specified in the source node, the separate Type node can be eliminated. Functionally, the result is the same.

This stream also has Table and Analysis nodes that will be used to view the scoring results after the model nugget has been created and added to the stream.

The Statistics File source node reads data in IBM SPSS Statistics format from the *tree_credit.sav* data file, which is installed in the *Demos* folder. (A special variable named \$CLEO_DEMOS is used to reference this folder under the current IBM SPSS Modeler installation. This ensures the path will be valid regardless of the current installation folder or version.)

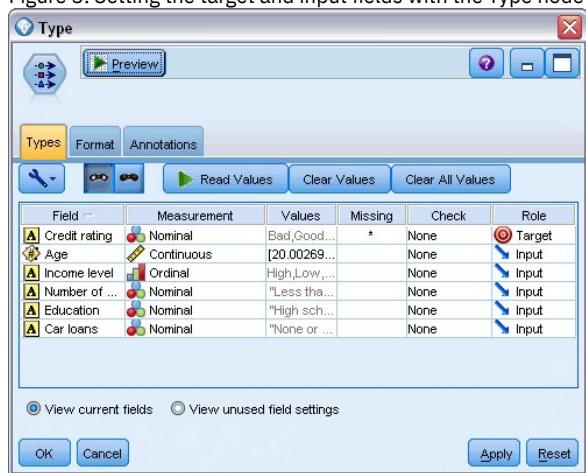
Figure 2. Reading data with a Statistics File source node



The Type node specifies the **measurement level** for each field. The measurement level is a category that indicates the type of data in the field. Our source data file uses three different measurement levels.

A **Continuous** field (such as the *Age* field) contains continuous numeric values, while a **Nominal** field (such as the *Credit rating* field) has two or more distinct values, for example *Bad*, *Good*, or *No credit history*. An **Ordinal** field (such as the *Income level* field) describes data with multiple distinct values that have an inherent order—in this case *Low*, *Medium* and *High*.

Figure 3. Setting the target and input fields with the Type node



For each field, the Type node also specifies a **role**, to indicate the part that each field plays in modeling. The role is set to *Target* for the field *Credit rating*, which is the field that indicates whether or not a given customer defaulted on the loan. This is the **target**, or the field for which we want to predict the value.

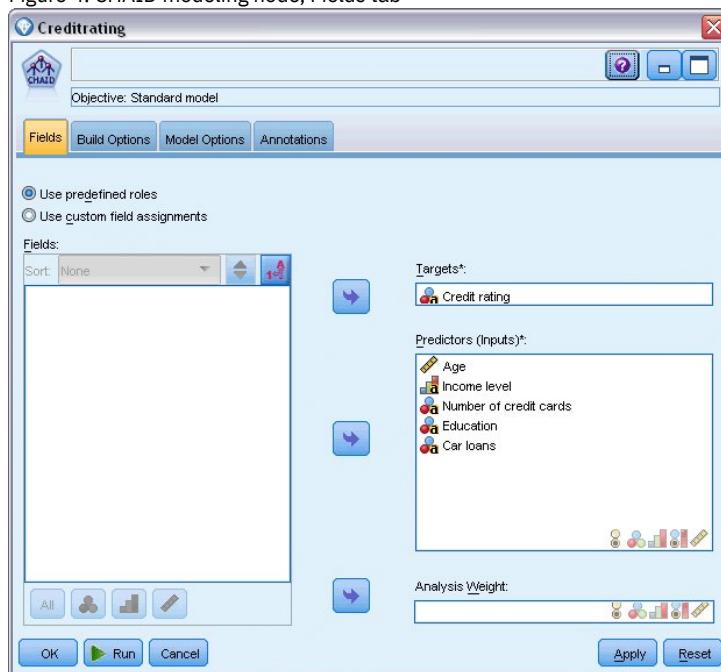
Role is set to *Input* for the other fields. Input fields are sometimes known as **predictors**, or fields whose values are used by the modeling algorithm to predict the value of the target field.

The CHAID modeling node generates the model.

On the Fields tab in the modeling node, the option *Use predefined roles* is selected, which means the target and inputs will be used as specified in the Type node. We could change the field roles at this point, but for this example we'll use them as they are.

1. Click the Build Options tab.

Figure 4. CHAID modeling node, Fields tab



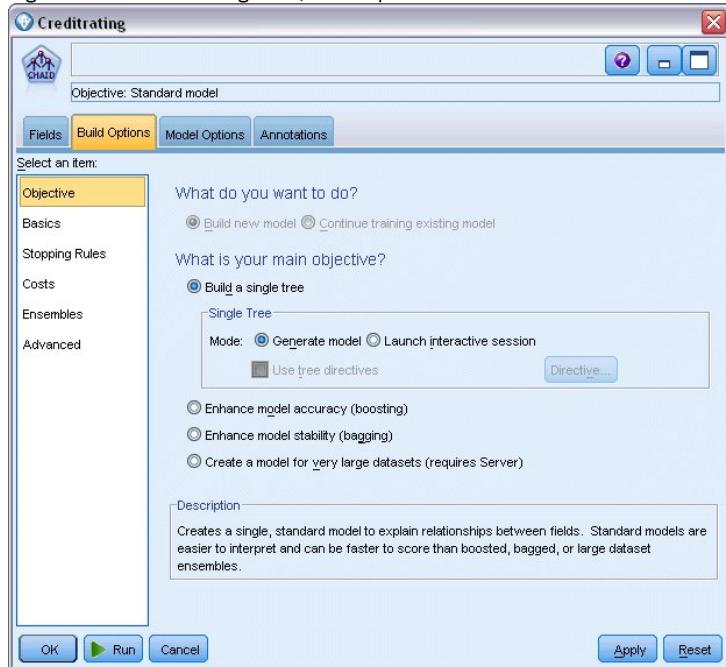
Here there are several options where we could specify the kind of model we want to build.

We want a brand-new model, so we'll use the default option *Build new model*.

We also just want a single, standard decision tree model without any enhancements, so we'll also leave the default objective option *Build a single tree*.

While we can optionally launch an interactive modeling session that allows us to fine-tune the model, this example simply generates a model using the default mode setting Generate model.

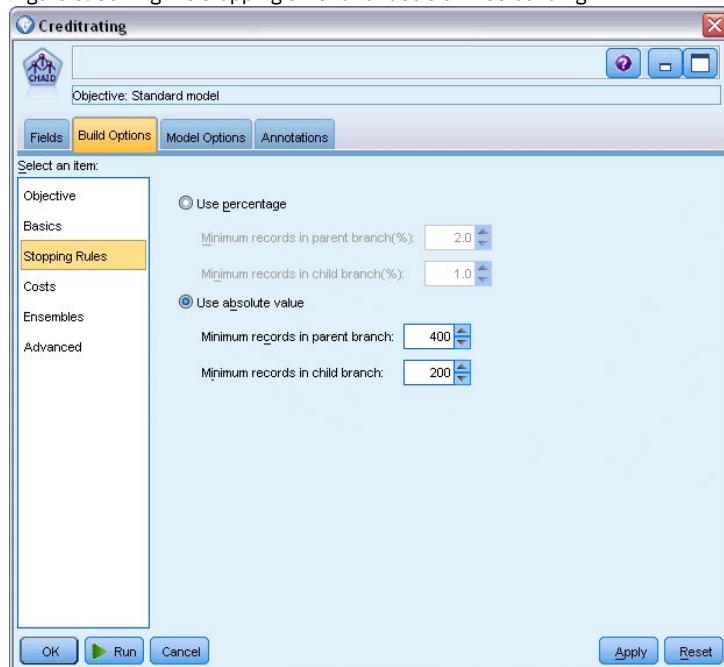
Figure 5. CHAID modeling node, Build Options tab



For this example, we want to keep the tree fairly simple, so we'll limit the tree growth by raising the minimum number of cases for parent and child nodes.

2. On the Build Options tab, select Stopping Rules from the navigator pane on the left.
3. Select the Use absolute value option.
4. Set Minimum records in parent branch to 400.
5. Set Minimum records in child branch to 200.

Figure 6. Setting the stopping criteria for decision tree building



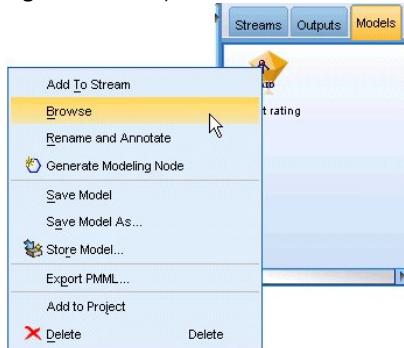
We can use all the other default options for this example, so click Run to create the model. (Alternatively, right-click on the node and choose Run from the context menu, or select the node and choose Run from the Tools menu.)

[Next](#)

Browsing the Model

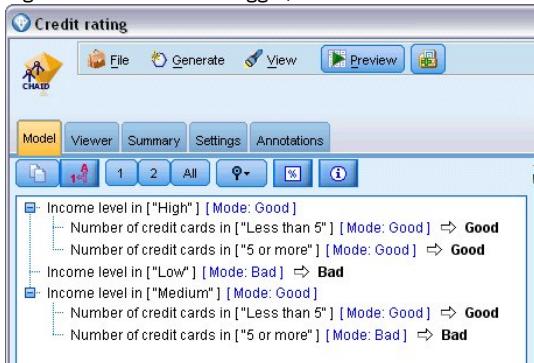
When execution completes, the model nugget is added to the Models palette in the upper right corner of the application window, and is also placed on the stream canvas with a link to the modeling node from which it was created. To view the model details, right-click on the model nugget and choose Browse (on the models palette) or Edit (on the canvas).

Figure 1. Models palette



In the case of the CHAID nugget, the Model tab displays the details in the form of a rule set--essentially a series of rules that can be used to assign individual records to child nodes based on the values of different input fields.

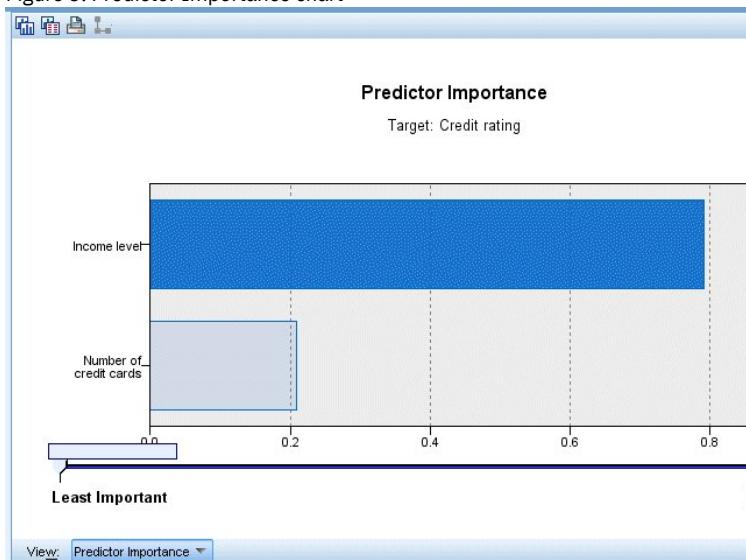
Figure 2. CHAID model nugget, rule set



For each decision tree terminal node--meaning those tree nodes that are not split further--a prediction of *Good* or *Bad* is returned. In each case the prediction is determined by the **mode**, or most common response, for records that fall within that node.

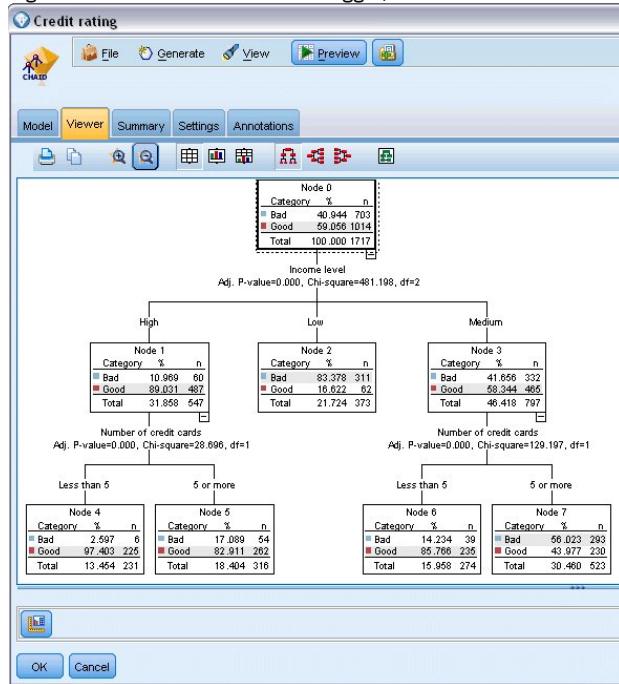
To the right of the rule set, the Model tab displays the Predictor Importance chart, which shows the relative importance of each predictor in estimating the model. From this we can see that *Income level* is easily the most significant in this case, and that the only other significant factor is *Number of credit cards*.

Figure 3. Predictor Importance chart



The Viewer tab in the model nugget displays the same model in the form of a tree, with a node at each decision point. Use the Zoom controls on the toolbar to zoom in on a specific node or zoom out to see the more of the tree.

Figure 4. Viewer tab in the model nugget, with zoom out selected



Looking at the upper part of the tree, the first node (Node 0) gives us a summary for all the records in the data set. Just over 40% of the cases in the data set are classified as a bad risk. This is quite a high proportion, so let's see if the tree can give us any clues as to what factors might be responsible.

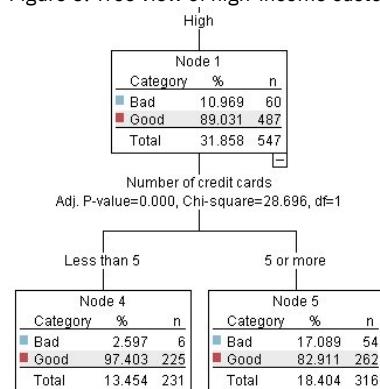
We can see that the first split is by *Income level*. Records where the income level is in the *Low* category are assigned to Node 2, and it's no surprise to see that this category contains the highest percentage of loan defaulters. Clearly lending to customers in this category carries a high risk.

However, 16% of the customers in this category actually *didn't* default, so the prediction won't always be correct. No model can feasibly predict every response, but a good model should allow us to predict the *most likely* response for each record based on the available data.

In the same way, if we look at the high income customers (Node 1), we see that the vast majority (89%) are a good risk. But more than 1 in 10 of these customers has also defaulted. Can we refine our lending criteria to minimize the risk here?

Notice how the model has divided these customers into two sub-categories (Nodes 4 and 5), based on the number of credit cards held. For high-income customers, if we lend only to those with fewer than 5 credit cards, we can increase our success rate from 89% to 97%--an even more satisfactory outcome.

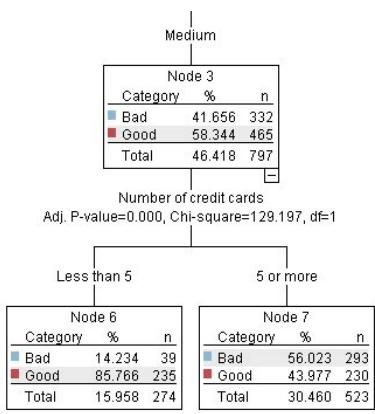
Figure 5. Tree view of high-income customers



But what about those customers in the *Medium* income category (Node 3)? They're much more evenly divided between Good and Bad ratings.

Again, the sub-categories (Nodes 6 and 7 in this case) can help us. This time, lending only to those medium-income customers with fewer than 5 credit cards increases the percentage of Good ratings from 58% to 85%, a significant improvement.

Figure 6. Tree view of medium-income customers



So, we've learnt that every record that is input to this model will be assigned to a specific node, and assigned a prediction of *Good* or *Bad* based on the most common response for that node.

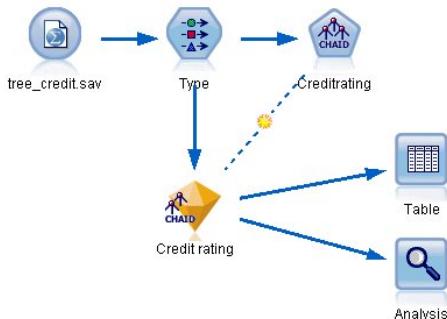
This process of assigning predictions to individual records is known as **scoring**. By scoring the same records used to estimate the model, we can evaluate how accurately it performs on the training data—the data for which we know the outcome. Let's look at how to do this.

[Next](#)

Evaluating the Model

We've been browsing the model to understand how scoring works. But to evaluate *how accurately* it works, we need to score some records and compare the responses predicted by the model to the actual results. We're going to score the same records that were used to estimate the model, allowing us to compare the observed and predicted responses.

Figure 1. Attaching the model nugget to output nodes for model evaluation



1. To see the scores or predictions, attach the Table node to the model nugget, double-click the Table node and click Run. The table displays the predicted scores in a field named *\$R-Credit rating*, which was created by the model. We can compare these values to the original *Credit rating* field that contains the actual responses.

By convention, the names of the fields generated during scoring are based on the target field, but with a standard prefix. Prefixes *\$G* and *\$GE* are generated by the Generalized Linear Model, *\$R* is the prefix used for the prediction generated by the CHAID model in this case, *\$RC* is for confidence values, *\$X* is typically generated by using an ensemble, and *\$XR*, *\$XS*, and *\$XF* are used as prefixes in cases where the target field is a Continuous, Categorical, Set, or Flag field, respectively. Different model types use different sets of prefixes. A **confidence value** is the model's own estimation, on a scale from 0.0 to 1.0, of how accurate each predicted value is.

Figure 2. Table showing generated scores and confidence values

Number of credit cards	Education	Car loans	\$R-Credit rating	\$RC-Credit rating
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	High school	More than 2	Bad	0.832
5 or more	College	None or 1	Bad	0.832
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	High school	More than 2	Bad	0.832
5 or more	High school	More than 2	Bad	0.832
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.832
5 or more	High school	More than 2	Bad	0.832
5 or more	High school	More than 2	Bad	0.560
5 or more	College	None or 1	Bad	0.832
5 or more	High school	More than 2	Bad	0.832
5 or more	College	More than 2	Bad	0.832
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Good	0.827

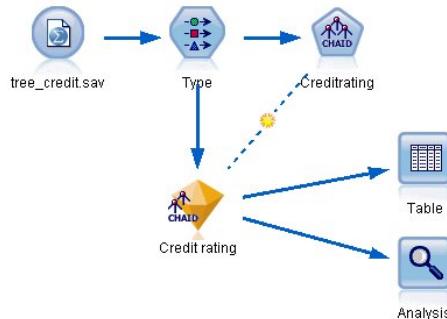
As expected, the predicted value matches the actual responses for many records but not all. The reason for this is that each CHAID terminal node has a mix of responses. The prediction matches the *most common* one, but will be wrong for all the others in that node. (Recall the 16% minority of low-income customers who did not default.)

To avoid this, we could continue splitting the tree into smaller and smaller branches, until every node was 100% pure—all *Good* or *Bad* with no mixed responses. But such a model would be extremely complicated and would probably not generalize well to other datasets.

To find out exactly how many predictions are correct, we could read through the table and tally the number of records where the value of the predicted field *\$R-Credit rating* matches the value of *Credit rating*. Fortunately, there's a much easier way—we can use an Analysis node, which does this automatically.

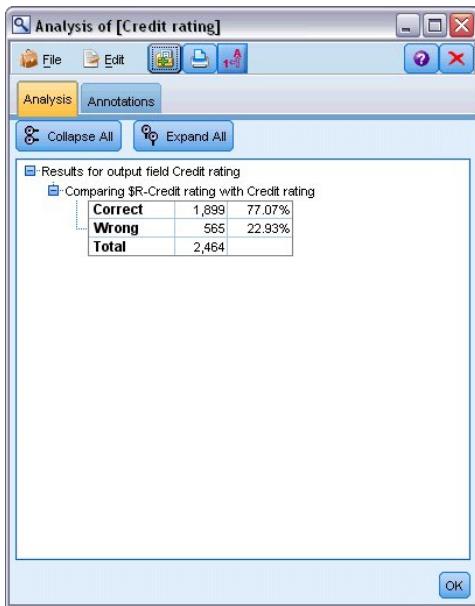
2. Connect the model nugget to the Analysis node.
3. Double-click the Analysis node and click Run.

Figure 3. Attaching an Analysis node



The analysis shows that for 1899 out of 2464 records--over 77%--the value predicted by the model matched the actual response.

Figure 4. Analysis results comparing observed and predicted responses



This result is limited by the fact that the records being scored are the same ones used to estimate the model. In a real situation, you could use a Partition node to split the data into separate samples for training and evaluation.

By using one sample partition to generate the model and another sample to test it, you can get a much better indication of how well it will generalize to other datasets.

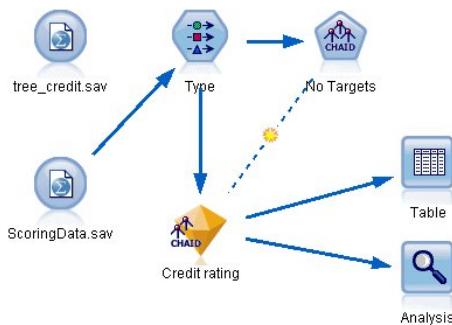
The Analysis node allows us to test the model against records for which we already know the actual result. The next stage illustrates how we can use the model to score records for which we don't know the outcome. For example, this might include people who are not currently customers of the bank, but who are prospective targets for a promotional mailing.

[Next](#)

Scoring records

Earlier, we scored the same records used to estimate the model in order to evaluate how accurate the model was. Now we're going to see how to score a different set of records from the ones used to create the model. This is the goal of modeling with a target field: Study records for which you know the outcome, to identify patterns that will allow you to predict outcomes you don't yet know.

Figure 1. Attaching new data for scoring



You could update the Statistics File source node to point to a different data file, or you could add a new source node that reads in the data you want to score. Either way, the new dataset must contain the same input fields used by the model (*Age*, *Income level*, *Education* and so on) but not the target field *Credit rating*.

Alternatively, you could add the model nugget to any stream that includes the expected input fields. Whether read from a file or a database, the source type doesn't matter as long as the field names and types match those used by the model.

You could also save the model nugget as a separate file, or export the model in PMML format for use with other applications that support this format, or store the model in an IBM® SPSS® Collaboration and Deployment Services repository, which offers enterprise-wide deployment, scoring, and management of models.

Regardless of the infrastructure used, the model itself works in the same way.

Summary

This example demonstrates the basic steps for creating, evaluating, and scoring a model.

- The modeling node estimates the model by studying records for which the outcome is known, and creates a model nugget. This is sometimes referred to as training the model.
- The model nugget can be added to any stream with the expected fields to score records. By scoring the records for which you already know the outcome (such as existing customers), you can evaluate how well it performs.
- Once you are satisfied that the model performs acceptably well, you can score new data (such as prospective customers) to predict how they will respond.
- The data used to train or estimate the model may be referred to as the analytical or historical data; the scoring data may also be referred to as the operational data.

Automated Modeling for a Flag Target

- [Modeling Customer Response \(Auto Classifier\)](#)
- [Historical Data](#)
- [Building the Stream](#)
- [Generating and Comparing Models](#)
- [Summary](#)

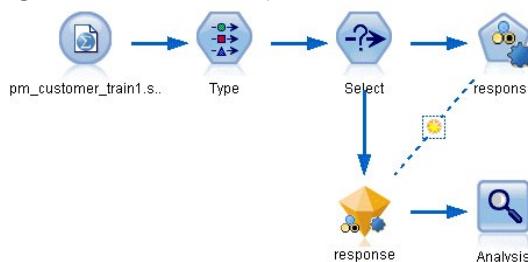
Modeling Customer Response (Auto Classifier)

The Auto Classifier node enables you to automatically create and compare a number of different models for either flag (such as whether or not a given customer is likely to default on a loan or respond to a particular offer) or nominal (set) targets. In this example we'll search for a flag (yes or no) outcome. Within a relatively simple stream, the node generates and ranks a set of candidate models, chooses the ones that perform the best, and combines them into a single aggregated (Ensembled) model. This approach combines the ease of automation with the benefits of combining multiple models, which often yield more accurate predictions than can be gained from any one model.

This example is based on a fictional company that wants to achieve more profitable results by matching the right offer to each customer.

This approach stresses the benefits of automation. For a similar example that uses a continuous (numeric range) target, see [Property Values \(Auto Numeric\)](#).

Figure 1. Auto Classifier sample stream



This example uses the stream *pm_binaryclassifier.str*, installed in the Demo folder under *streams*. The data file used is *pm_customer_train1.sav*. See the topic [Historical Data](#) for more information.

[Next](#)

Historical Data

The file *pm_customer_train1.sav* has historical data tracking the offers made to specific customers in past campaigns, as indicated by the value of the *campaign* field. The largest number of records fall under the *Premium account* campaign.

The values of the *campaign* field are actually coded as integers in the data (for example 2 = *Premium account*). Later, you'll define labels for these values that you can use to give more meaningful output.

Figure 1. Data about previous promotions

The screenshot shows a software interface titled "Table (31 fields, 21,927 records)". The window has a toolbar with icons for File, Edit, Generate, and various data manipulation tools. Below the toolbar is a menu bar with "Table" and "Annotations". The main area is a grid of data rows. The first few rows are:

	customer_id	campaign	response	response_date	purchase	purchase_date	product_id	Rowid
1	7	2	0	\$null\$	0	\$null\$	\$null\$	1
2	13	2	0	\$null\$	0	\$null\$	\$null\$	2
3	15	2	0	\$null\$	0	\$null\$	\$null\$	3
4	16	2	1	2006-07-05 00:00:00	0	\$null\$	183	761
5	23	2	0	\$null\$	0	\$null\$	\$null\$	4
6	24	2	0	\$null\$	0	\$null\$	\$null\$	5
7	30	2	0	\$null\$	0	\$null\$	\$null\$	6
8	30	3	0	\$null\$	0	\$null\$	\$null\$	7
9	33	2	0	\$null\$	0	\$null\$	\$null\$	8
10	42	3	0	\$null\$	0	\$null\$	\$null\$	9
11	42	2	0	\$null\$	0	\$null\$	\$null\$	10
12	52	2	0	\$null\$	0	\$null\$	\$null\$	11
13	57	2	0	\$null\$	0	\$null\$	\$null\$	12
14	63	2	1	2006-07-14 00:00:00	0	\$null\$	183	1501
15	74	2	0	\$null\$	0	\$null\$	\$null\$	13
16	74	3	0	\$null\$	0	\$null\$	\$null\$	14
17	75	2	0	\$null\$	0	\$null\$	\$null\$	15
18	82	2	0	\$null\$	0	\$null\$	\$null\$	16
19	89	3	0	\$null\$	0	\$null\$	\$null\$	17
20	89	2	0	\$null\$	0	\$null\$	\$null\$	18

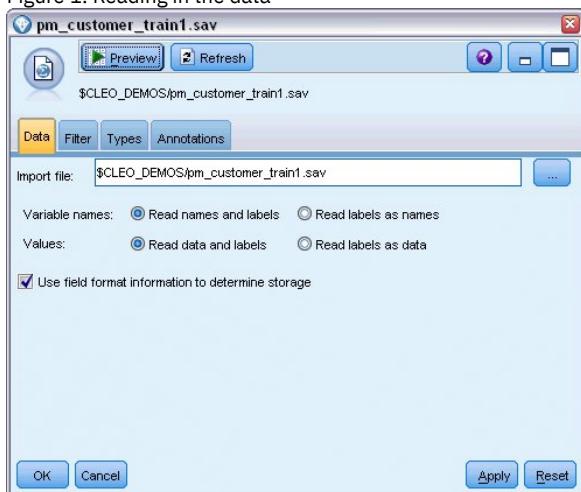
The file also includes a *response* field that indicates whether the offer was accepted (0 = no, and 1 = yes). This will be the **target field**, or value, that you want to predict. A number of fields containing demographic and financial information about each customer are also included. These can be used to build or "train" a model that predicts response rates for individuals or groups based on characteristics such as income, age, or number of transactions per month.

[Next](#)

Building the Stream

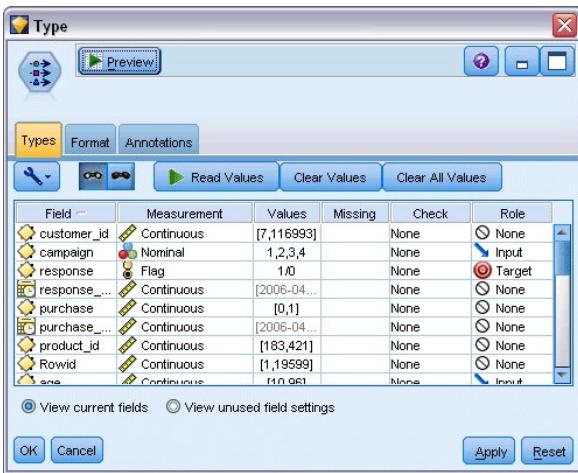
1. Add a Statistics File source node pointing to *pm_customer_train1.sav*, located in the *Demos* folder of your IBM® SPSS® Modeler installation. (You can specify **\$CLEO_DEMOS/** in the file path as a shortcut to reference this folder. Note that a forward slash—rather than a backslash—must be used in the path, as shown.)

Figure 1. Reading in the data



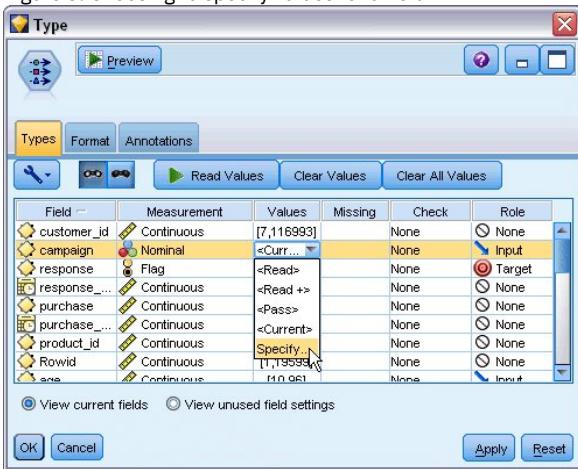
2. Add a Type node, and select *response* as the target field (Role = Target). Set the Measurement for this field to Flag.

Figure 2. Setting the measurement level and role



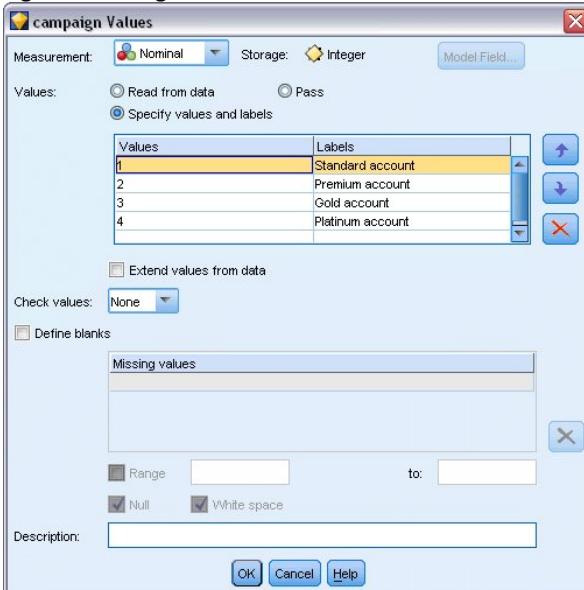
- Set the role to None for the following fields: *customer_id*, *campaign*, *response_date*, *purchase*, *purchase_date*, *product_id*, *Rowid*, and *X_random*. These fields will be ignored when you are building the model.
 - Click the Read Values button in the Type node to make sure that values are instantiated.
- As we saw earlier, our source data includes information about four different campaigns, each targeted to a different type of customer account. These campaigns are coded as integers in the data, so to make it easier to remember which account type each integer represents, let's define labels for each one.

Figure 3. Choosing to specify values for a field



- On the row for the campaign field, click the entry in the Values column.
- Choose Specify from the drop-down list.

Figure 4. Defining labels for the field values



- In the Labels column, type the labels as shown for each of the four values of the campaign field.
- Click OK.

Now you can display the labels in output windows instead of the integers.

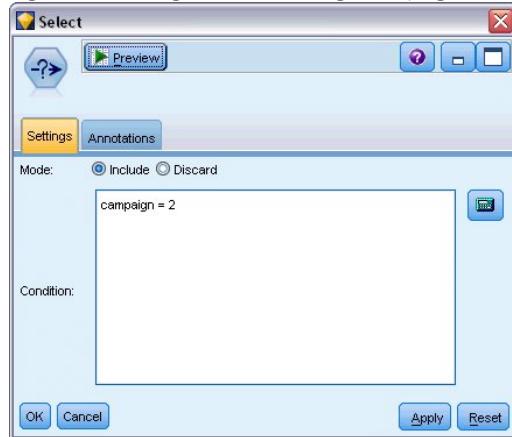
Figure 5. Displaying the field value labels

	customer_id	campaign	response	response_date	purchase	purchase_date	product_id
1	7	Premium account	0	\$null\$	0	\$null\$	1
2	13	Premium account	0	\$null\$	0	\$null\$	2
3	15	Premium account	0	\$null\$	0	\$null\$	3
4	16	Premium account	1	2006-07-05 00:00:00	0	\$null\$	183
5	23	Premium account	0	\$null\$	0	\$null\$	4
6	24	Premium account	0	\$null\$	0	\$null\$	5
7	30	Premium account	0	\$null\$	0	\$null\$	6
8	30	Gold account	0	\$null\$	0	\$null\$	7
9	33	Premium account	0	\$null\$	0	\$null\$	8
10	42	Gold account	0	\$null\$	0	\$null\$	9
11	42	Premium account	0	\$null\$	0	\$null\$	10
12	52	Premium account	0	\$null\$	0	\$null\$	11
13	57	Premium account	0	\$null\$	0	\$null\$	12
14	63	Premium account	1	2006-07-14 00:00:00	0	\$null\$	183
15	74	Premium account	0	\$null\$	0	\$null\$	13
16	74	Gold account	0	\$null\$	0	\$null\$	14
17	75	Premium account	0	\$null\$	0	\$null\$	15
18	82	Premium account	0	\$null\$	0	\$null\$	16
19	89	Gold account	0	\$null\$	0	\$null\$	17
20	89	Premium account	0	\$null\$	0	\$null\$	18

- Attach a Table node to the Type node.
- Open the Table node and click Run.
- On the output window, click the Display field and value labels toolbar button to display the labels.
- Click OK to close the output window.

Although the data includes information about four different campaigns, you will focus the analysis on one campaign at a time. Since the largest number of records fall under the Premium account campaign (coded *campaign*=2 in the data), you can use a Select node to include only these records in the stream.

Figure 6. Selecting records for a single campaign

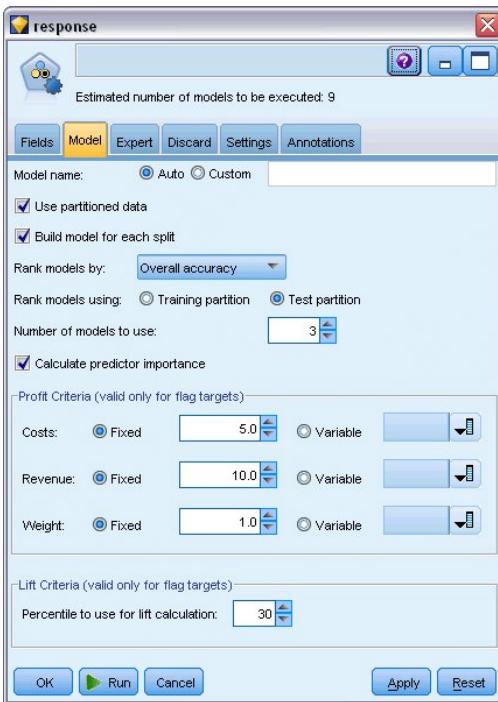


[Next](#)

Generating and Comparing Models

- Attach an Auto Classifier node, and select Overall Accuracy as the metric used to rank models.
- Set the Number of models to use to 3. This means that the three best models will be built when you execute the node.

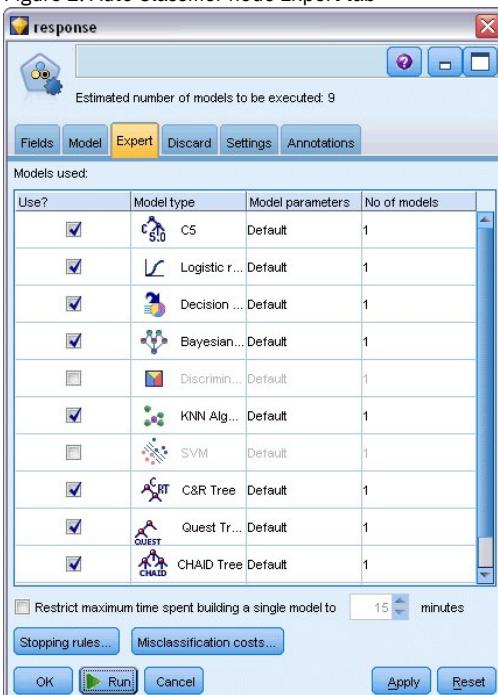
Figure 1. Auto Classifier node Model tab



On the Expert tab you can choose from up to 11 different model algorithms.

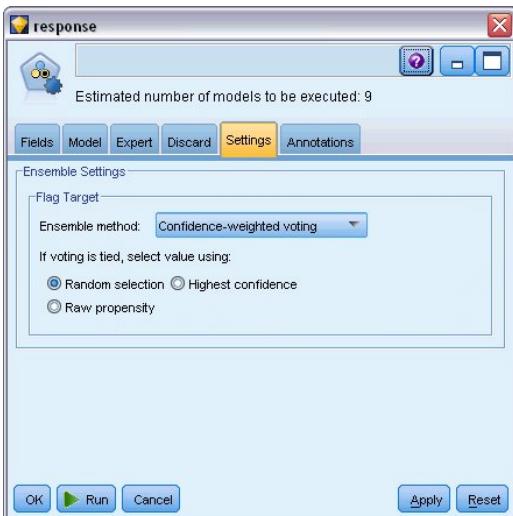
3. Deselect the Discriminant and SVM model types. (These models take longer to train on these data, so deselecting them will speed up the example. If you don't mind waiting, feel free to leave them selected.)
- Because you set Number of models to use to 3 on the Model tab, the node will calculate the accuracy of the remaining nine algorithms and build a single model nugget containing the three most accurate.

Figure 2. Auto Classifier node Expert tab



4. On the Settings tab, for the ensemble method, select Confidence-weighted voting. This determines how a single aggregated score is produced for each record.
- With simple voting, if two out of three models predict yes, then yes wins by a vote of 2 to 1. In the case of confidence-weighted voting, the votes are weighted based on the confidence value for each prediction. Thus, if one model predicts no with a higher confidence than the two yes predictions combined, then no wins.

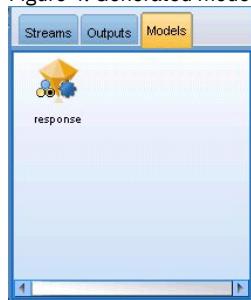
Figure 3. Auto Classifier node: Settings tab



5. Click Run.

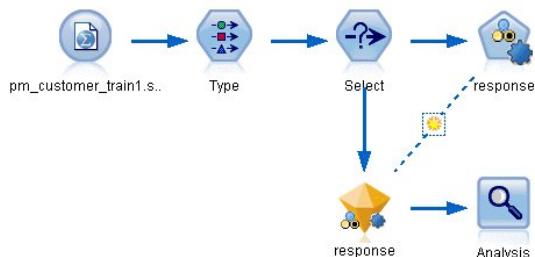
After a few minutes, the generated model nugget is built and placed on the canvas, and on the Models palette in the upper right corner of the window. You can browse the model nugget, or save or deploy it in a number of other ways.

Figure 4. Generated model displayed in the palette



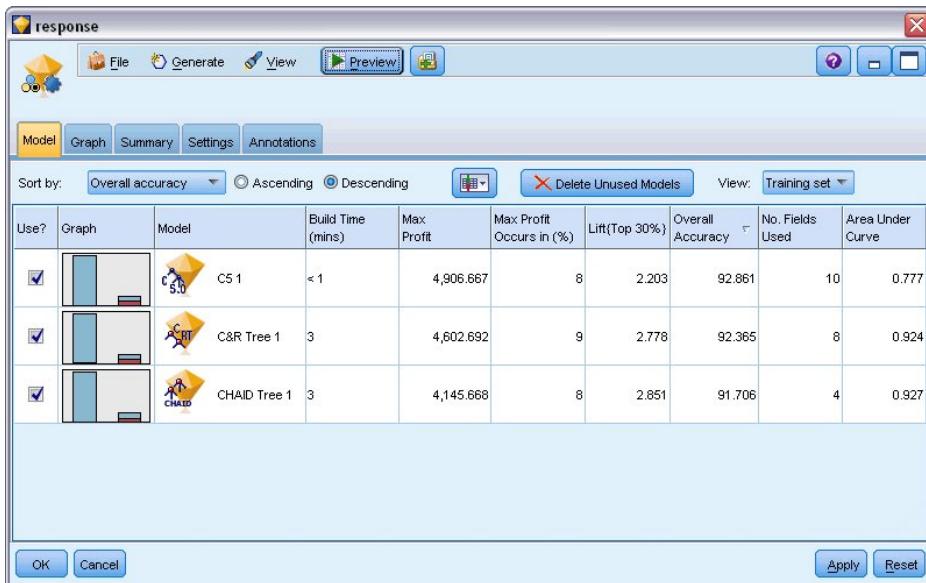
Open the model nugget; it lists details about each of the models created during the run. (In a real situation, in which hundreds of models may be created on a large dataset, this could take many hours.)

Figure 5. Auto Classifier stream with model nugget



If you want to explore any of the individual models further, you can double-click on a model nugget icon in the Model column to drill down and browse the individual model results; from there you can generate modeling nodes, model nuggets, or evaluation charts. In the Graph column, you can double-click on a thumbnail to generate a full-sized graph.

Figure 6. Auto Classifier results



By default, models are sorted based on overall accuracy, because this was the measure you selected on the Auto Classifier node Model tab. The C51 model ranks best by this measure, but the C&R Tree and CHAID models are nearly as accurate.

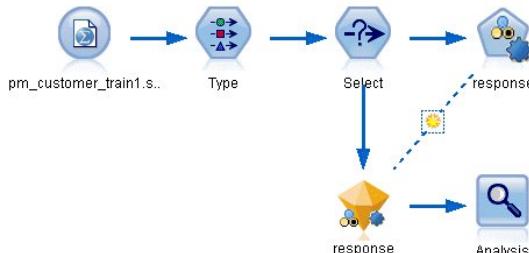
You can sort on a different column by clicking the header for that column, or you can choose the desired measure from the Sort by drop-down list on the toolbar.

Based on these results, you decide to use all three of these most accurate models. By combining predictions from multiple models, limitations in individual models may be avoided, resulting in a higher overall accuracy.

In the Use? column, select the C51, C&R Tree, and CHAID models.

Attach an Analysis node (Output palette) after the model nugget. Right-click on the Analysis node and choose Run to run the stream.

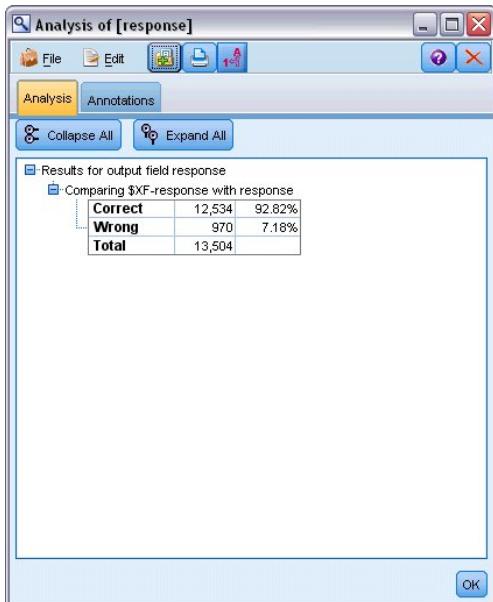
Figure 7. Auto Classifier sample stream



The aggregated score generated by the ensembled model is shown in a field named `$XF-response`. When measured against the training data, the predicted value matches the actual response (as recorded in the original `response` field) with an overall accuracy of 92.82%.

While not quite as accurate as the best of the three individual models in this case (92.86% for C51), the difference is too small to be meaningful. In general terms, an ensembled model will typically be more likely to perform well when applied to datasets other than the training data.

Figure 8. Analysis of the three ensembled models



Summary

To sum up, you used the Auto Classifier node to compare a number of different models, used the three most accurate models and added them to the stream within an ensembled Auto Classifier model nugget.

- Based on overall accuracy, the C51, C&R Tree, and CHAID models performed best on the training data.
- The ensembled model performed nearly as well as the best of the individual models and may perform better when applied to other datasets. If your goal is to automate the process as much as possible, this approach allows you to obtain a robust model under most circumstances without having to dig deeply into the specifics of any one model.

Automated Modeling for a Continuous Target

- [Property Values \(Auto Numeric\)](#)
- [Training Data](#)
- [Building the Stream](#)
- [Comparing the Models](#)
- [Summary](#)

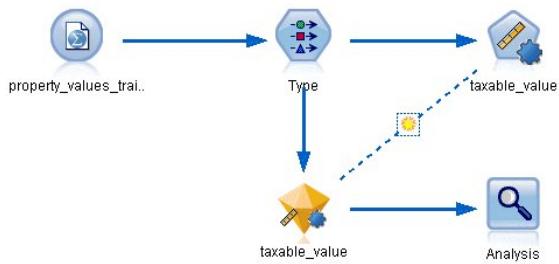
Property Values (Auto Numeric)

The Auto Numeric node enables you to automatically create and compare different models for continuous (numeric range) outcomes, such as predicting the taxable value of a property. With a single node, you can estimate and compare a set of candidate models and generate a subset of models for further analysis. The node works in the same manner as the Auto Classifier node, but for continuous rather than flag or nominal targets.

The node combines the best of the candidate models into a single aggregated (Ensembled) model nugget. This approach combines the ease of automation with the benefits of combining multiple models, which often yield more accurate predictions than can be gained from any one model.

This example focuses on a fictional municipality responsible for adjusting and assessing real estate taxes. To do this more accurately, they will build a model that predicts property values based on building type, neighborhood, size, and other known factors.

Figure 1. Auto Numeric sample stream



This example uses the stream *property_values_numericpredictor.str*, installed in the Demos folder under streams. The data file used is *property_values_train.sav*. See the topic [Demos Folder](#) for more information.

[Next](#)

Training Data

The data file includes a field named *taxable_value*, which is the **target field**, or value, that you want to predict. The other fields contain information such as neighborhood, building type, and interior volume and may be used as predictors.

Field name	Label
<i>property_id</i>	Property ID
<i>neighborhood</i>	Area within the city
<i>building_type</i>	Type of building
<i>year_built</i>	Year built
<i>volume_interior</i>	Volume of interior
<i>volume_other</i>	Volume of garage and extra buildings
<i>lot_size</i>	Lot size
<i>taxable_value</i>	Taxable value

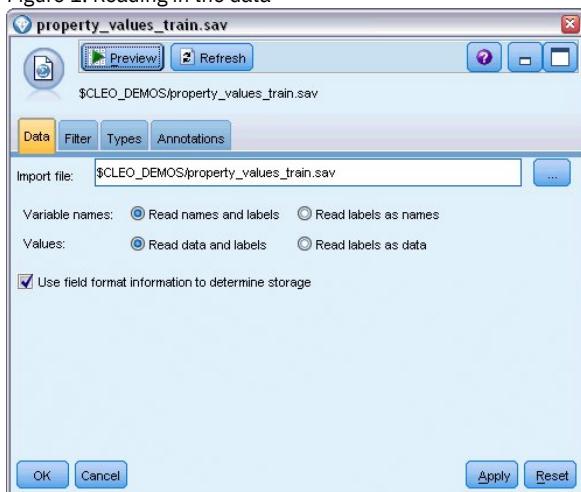
A scoring data file named *property_values_score.sav* is also included in the Demos folder. It contains the same fields but without the *taxable_value* field. After training models using a dataset where the taxable value is known, you can score records where this value is not yet known.

[Next](#)

Building the Stream

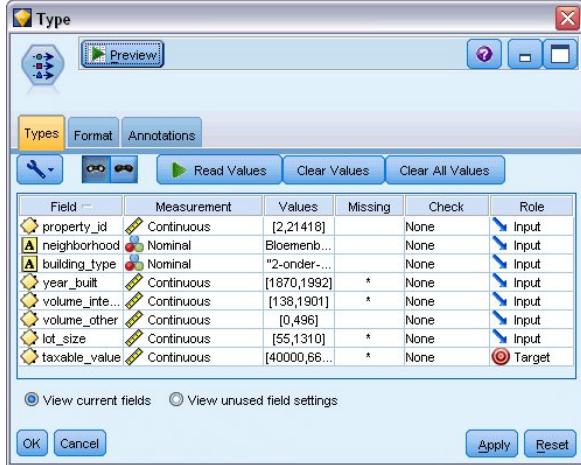
1. Add a Statistics File source node pointing to *property_values_train.sav*, located in the Demos folder of your IBM® SPSS® Modeler installation. (You can specify **\$CLEO_DEMOS/** in the file path as a shortcut to reference this folder. Note that a forward slash—rather than a backslash—must be used in the path, as shown.)

Figure 1. Reading in the data



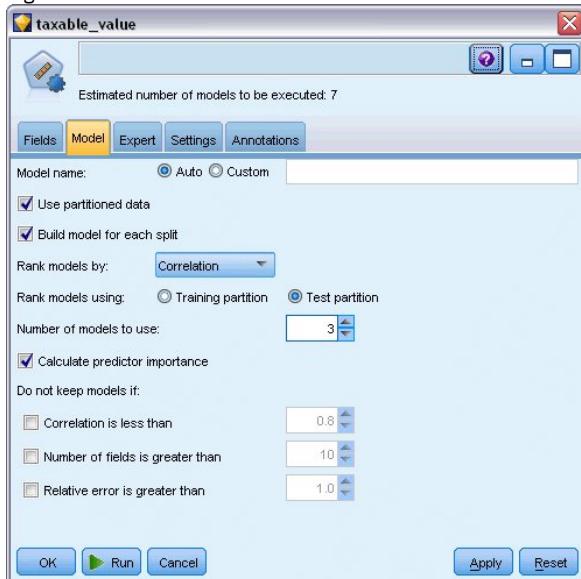
2. Add a Type node, and select *taxable_value* as the target field (Role = Target). Role should be set to Input for all other fields, indicating that they will be used as predictors.

Figure 2. Setting the target field



3. Attach an Auto Numeric node, and select Correlation as the metric used to rank models.
4. Set the Number of models to use to 3. This means that the three best models will be built when you execute the node.

Figure 3. Auto Numeric node Model tab



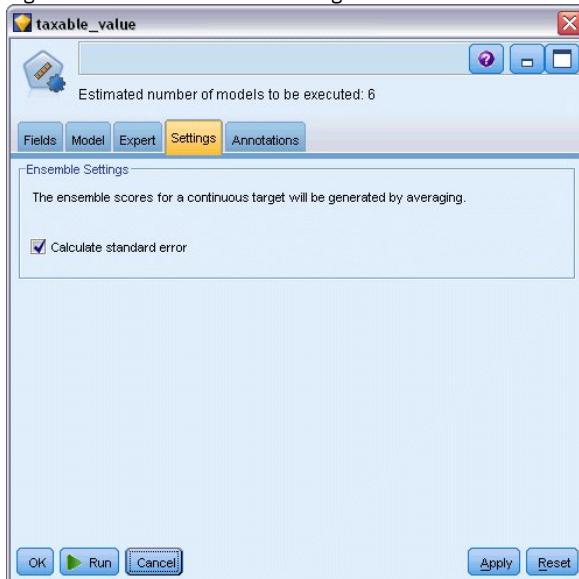
5. On the Expert tab, leave the default settings in place; the node will estimate a single model for each algorithm, for a total of seven models. (Alternatively, you can modify these settings to compare multiple variants for each model type.)
- Because you set Number of models to use to 3 on the Model tab, the node will calculate the accuracy of the seven algorithms and build a single model nugget containing the three most accurate.

Figure 4. Auto Numeric node Expert tab



6. On the Settings tab, leave the default settings in place. Since this is a continuous target, the ensemble score is generated by averaging the scores for the individual models.

Figure 5. Auto Numeric node Settings tab



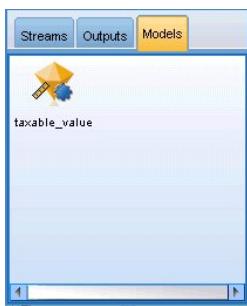
[Next](#)

Comparing the Models

1. Click the Run button.

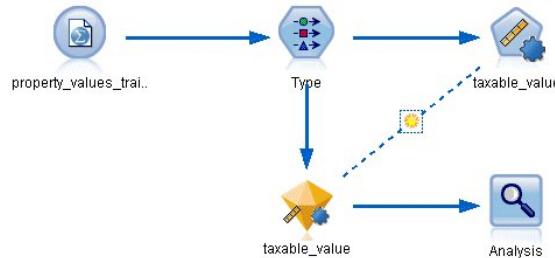
The model nugget is built and placed on the canvas, and also on the Models palette in the upper right corner of the window. You can browse the nugget, or save or deploy it in a number of other ways.

Figure 1. Model displayed in the Models palette



Open the model nugget; it lists details about each of the models created during the run. (In a real situation, in which hundreds of models are estimated on a large dataset, this could take many hours.)

Figure 2. Auto Numeric sample stream with model nugget



If you want to explore any of the individual models further, you can double-click on a model nugget icon in the Model column to drill down and browse the individual model results; from there you can generate modeling nodes, model nuggets, or evaluation charts.

Figure 3. Auto Numeric results

Use?	Graph	Model	Build Time (mins)	Correlation r	No. Fields Used	Relative Error
<input checked="" type="checkbox"/>		Generalized Linear 1	< 1	0.915	7	0.162
<input checked="" type="checkbox"/>		Regression 1	< 1	0.9	5	0.19
<input checked="" type="checkbox"/>		CHAID Tree 1	< 1	0.892	5	0.204

By default, models are sorted by correlation because this was the measure you selected in the Auto Numeric node. For purposes of ranking, the absolute value of the correlation is used, with values closer to 1 indicating a stronger relationship. The Generalized Linear model ranks best on this measure, but several others are nearly as accurate. The Generalized Linear model also has the lowest relative error.

You can sort on a different column by clicking the header for that column, or you can choose the desired measure from the Sort by list on the toolbar.

Each graph displays a plot of observed values against predicted values for the model, providing a quick visual indication of the correlation between them. For a good model, points should cluster along the diagonal, which is true for all the models in this example.

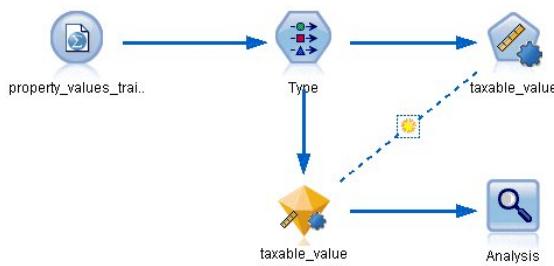
In the Graph column, you can double-click on a thumbnail to generate a full-sized graph.

Based on these results, you decide to use all three of these most accurate models. By combining predictions from multiple models, limitations in individual models may be avoided, resulting in a higher overall accuracy.

In the Use? column, ensure that all three models are selected.

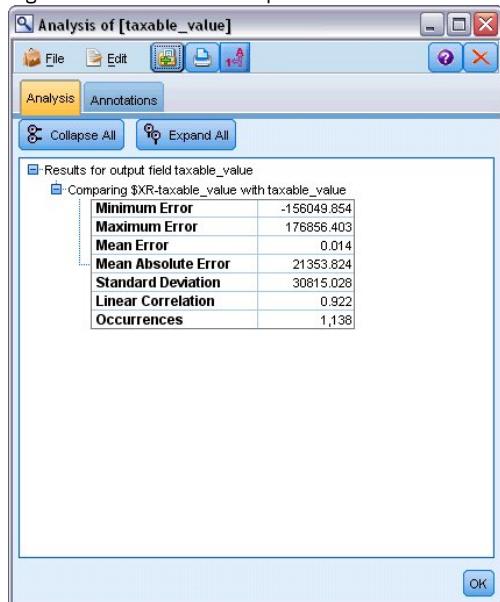
Attach an Analysis node (Output palette) after the model nugget. Right-click on the Analysis node and choose Run to run the stream.

Figure 4. Auto Numeric sample stream



The averaged score generated by the ensembled model is added in a field named \$XR-taxable_value, with a correlation of 0.922, which is higher than those of the three individual models. The ensemble scores also show a low mean absolute error and may perform better than any of the individual models when applied to other datasets.

Figure 5. Auto Numeric sample stream



[Next](#)

Summary

To sum up, you used the Auto Numeric node to compare a number of different models, selected the three most accurate models and added them to the stream within an ensembled Auto Numeric model nugget.

- Based on overall accuracy, the Generalized Linear, Regression, and CHAID models performed best on the training data.
- The ensembled model showed performance that was better than two of the individual models and may perform better when applied to other datasets. If your goal is to automate the process as much as possible, this approach allows you to obtain a robust model under most circumstances without having to dig deeply into the specifics of any one model.

Automated Data Preparation (ADP)

Preparing data for analysis is one of the most important steps in any data-mining project—and traditionally, one of the most time consuming. The Automated Data Preparation (ADP) node handles the task for you, analyzing your data and identifying fixes, screening out fields that are problematic or not likely to be useful, deriving new attributes when appropriate, and improving performance through intelligent screening techniques. You can use the node in fully automated fashion, allowing the node to choose and apply fixes, or you can preview the changes before they are made and accept or reject them as desired.

Using the ADP node enables you to make your data ready for data mining quickly and easily, without needing to have prior knowledge of the statistical concepts involved. If you run the node with the default settings, models will tend to build and score more quickly.

This example uses the stream named *ADP_basic_demo.str*, which references the data file named *telco.sav* to demonstrate the increased accuracy that may be found by using the default ADP node settings when building models. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *ADP_basic_demo.str* file is in the *streams* directory.

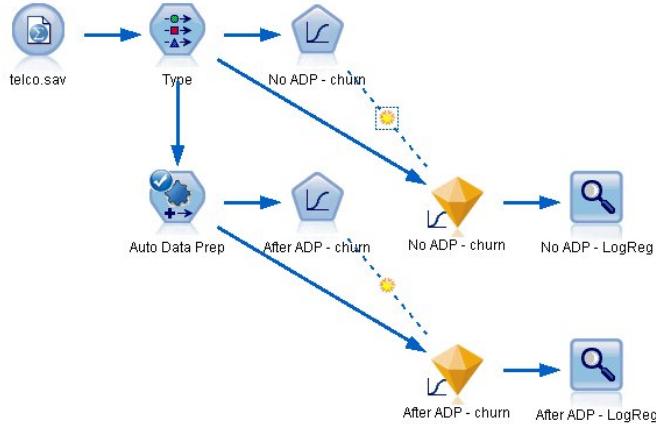
[Next](#)

- [Building the stream](#)
- [Comparing Model Accuracy](#)

Building the stream

1. To build the stream, add a Statistics File source node pointing to telco.sav located in the Demos directory of your IBM® SPSS® Modeler installation.

Figure 1. Building the stream



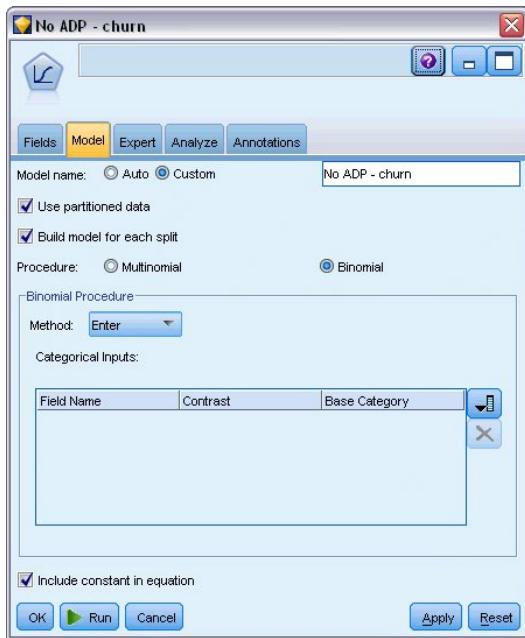
2. Attach a Type node to the source node, set the measurement level for the *churn* field to Flag, and set the role to Target. All other fields should have their role set to Input.

Figure 2. Selecting the target



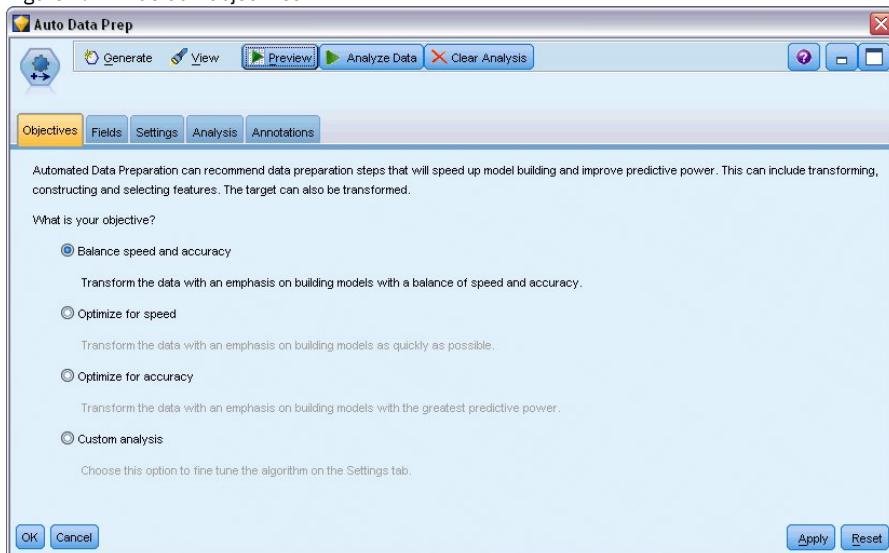
3. Attach a Logistic node to the Type node.
4. In the Logistic node, click the Model tab and select the Binomial procedure. In the *Model name* field, select Custom and enter *No ADP - churn*.

Figure 3. Choosing model options



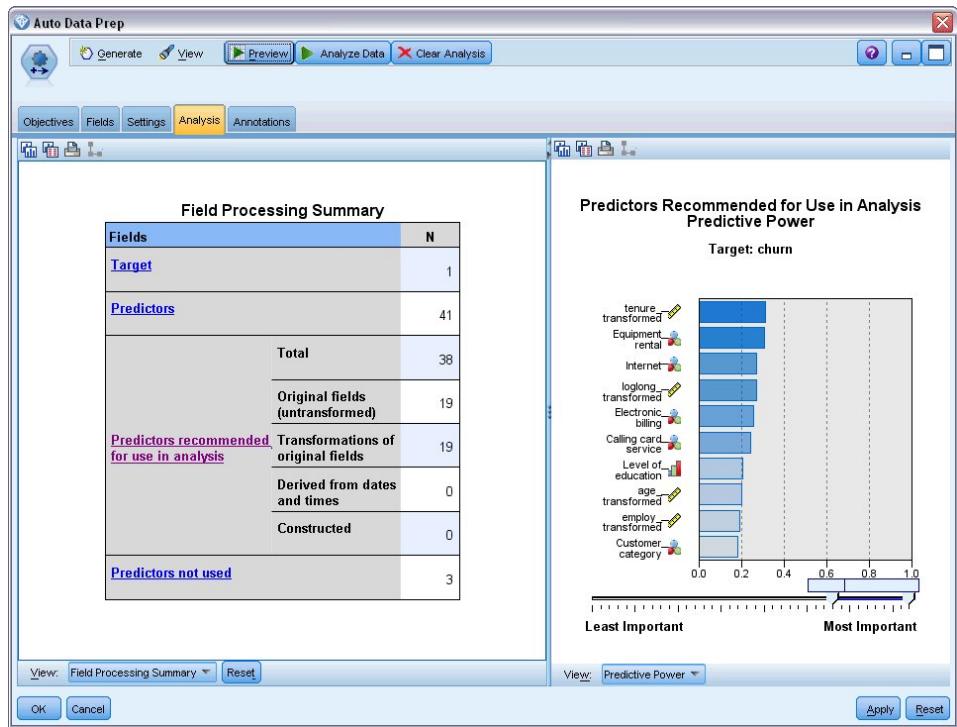
5. Attach an ADP node to the Type node. On the Objectives tab, leave the default settings in place to analyze and prepare your data by balancing both speed and accuracy.
6. At the top of the Objectives tab, click Analyze Data to analyze and process your data. Other options on the ADP node enable you to specify that you want to concentrate more on accuracy, more on the speed of processing, or to fine tune many of the data preparation processing steps.

Figure 4. ADP default objectives



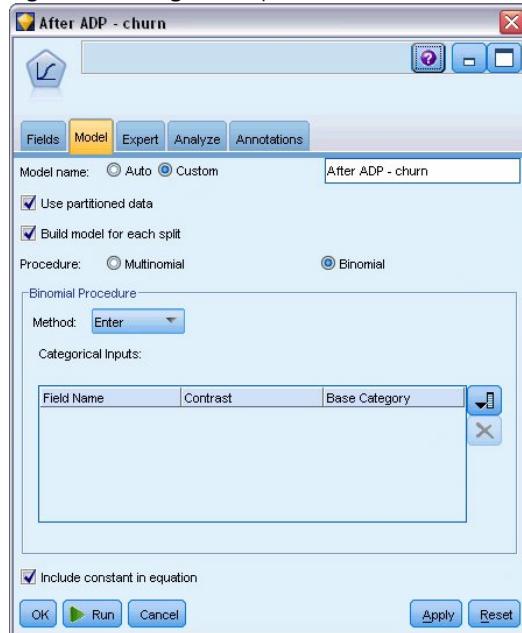
The results of the data processing are displayed on the Analysis tab. The Field Processing Summary shows that of the 41 data features brought in to the ADP node, 19 have been transformed to aid processing, and 3 have been discarded as unused.

Figure 5. Summary of data processing



7. Attach a Logistic node to the ADP node.
8. In the Logistic node, click the Model tab and select the Binomial procedure. In the *Modeling name* field, select Custom and enter *After ADP - churn*.

Figure 6. Choosing model options

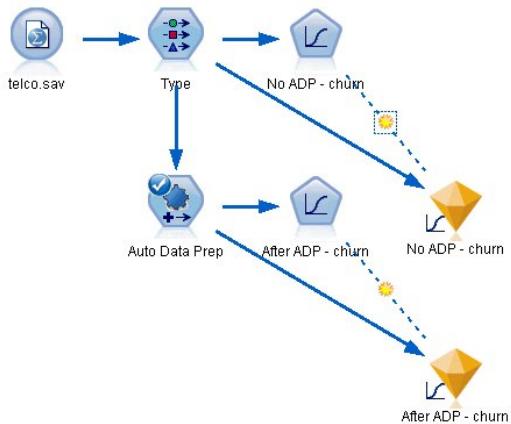


[Next](#)

Comparing Model Accuracy

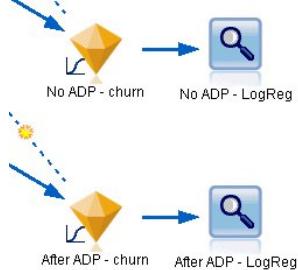
1. Run both Logistic nodes to create the model nuggets, which are added to the stream and to the Models palette in the upper-right corner.

Figure 1. Attaching the model nuggets



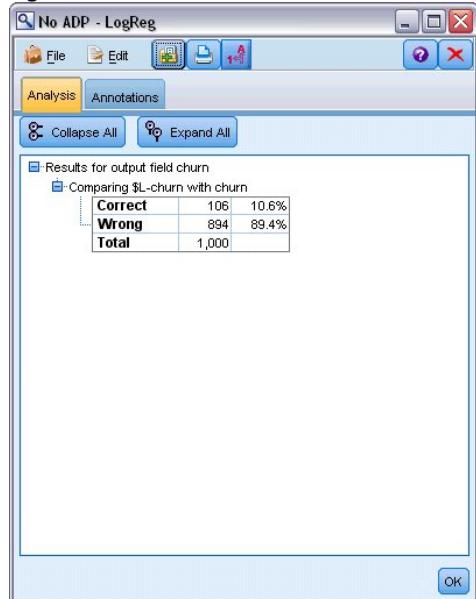
2. Attach Analysis nodes to the model nuggets and run the Analysis nodes using their default settings.

Figure 2. Attaching the Analysis nodes



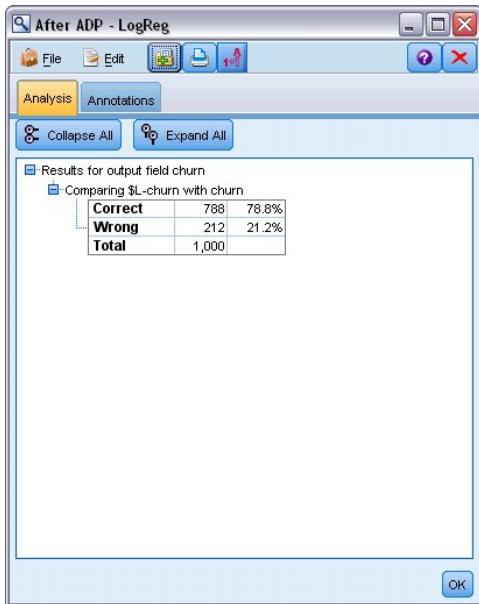
The Analysis of the non ADP-derived model shows that just running the data through the Logistic Regression node with its default settings gives a model with low accuracy - just 10.6%.

Figure 3. Non ADP-derived model results



The Analysis of the ADP-derived model shows that running the data through the default ADP settings, you have built a much more accurate model that is 78.8% correct.

Figure 4. ADP-derived model results



In summary, by just running the ADP node to fine tune the processing of your data, you were able to build a more accurate model with little direct data manipulation.

Obviously, if you are interested in proving or disproving a certain theory, or want to build specific models, you may find it beneficial to work directly with the model settings; however, for those with a reduced amount of time, or with a large amount of data to prepare, the ADP node may give you an advantage.

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the *IBM SPSS Modeler Algorithms Guide*, available from the *|Documentation* directory of the installation disk.

Note that the results in this example are based on the training data only. To assess how well models generalize to other data in the real world, you would use a Partition node to hold out a subset of records for purposes of testing and validation.

Preparing Data for Analysis (Data Audit)

The Data Audit node provides a comprehensive first look at the data you bring into IBM® SPSS® Modeler. Often used during the initial data exploration, the data audit report shows summary statistics as well as histograms and distribution graphs for each data field, and it allows you to specify treatments for missing values, outliers, and extreme values.

This example uses the stream named *telco_dataaudit.str*, which references the data file named *telco.sav*. These files are available from the *Demos* directory of any IBM SPSS Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *telco_dataaudit.str* file is in the *streams* directory.

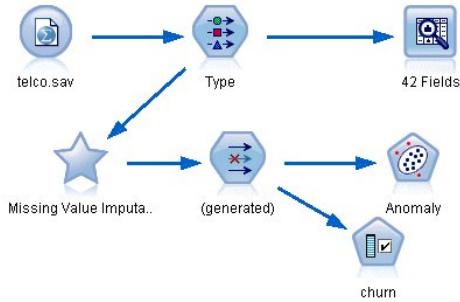
[Next](#)

- [Building the Stream](#)
- [Browsing Statistics and Charts](#)
- [Handling Outliers and Missing Values](#)

Building the Stream

1. To build the stream, add a Statistics File source node pointing to *telco.sav* located in the *Demos* directory of your IBM® SPSS® Modeler installation.

Figure 1. Building the stream



2. Add a Type node to define fields, and specify *churn* as the target field (Role = Target). Role should be set to Input for all of the other fields so that this is the only target.

Figure 2. Setting the target



3. Confirm that field measurement levels are defined correctly. For example, most fields with values 0 and 1 can be regarded as flags, but certain fields, such as gender, are more accurately viewed as a nominal field with two values.

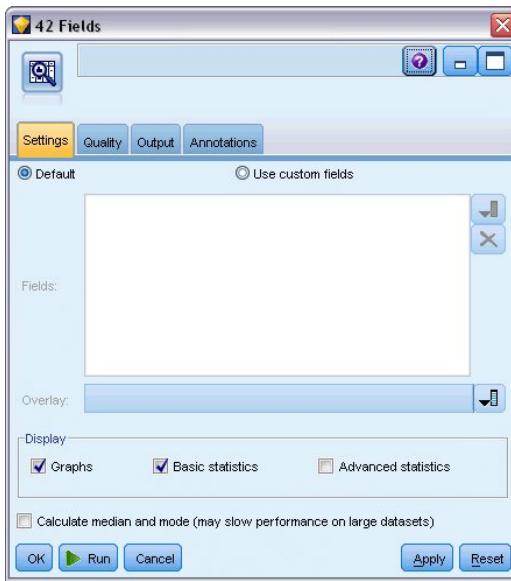
Figure 3. Setting measurement levels



Tip: To change properties for multiple fields with similar values (such as 0/1), click the *Values* column header to sort fields by that column, and use the Shift key to select all of the fields you want to change. You can then right-click on the selection to change the measurement level or other attributes for all selected fields.

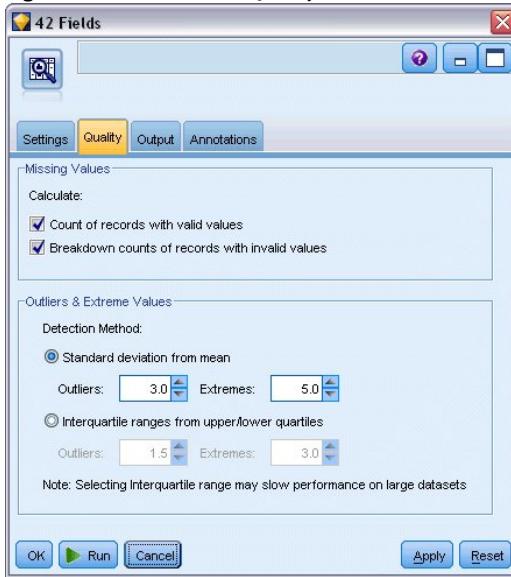
4. Attach a Data Audit node to the stream. On the Settings tab, leave the default settings in place to include all fields in the report. Since *churn* is the only target field defined in the Type node, it will automatically be used as an overlay.

Figure 4. Data Audit node, Settings tab



On the Quality tab, leave the default settings for detecting missing values, outliers, and extreme values in place, and click Run.

Figure 5. Data Audit node, Quality tab

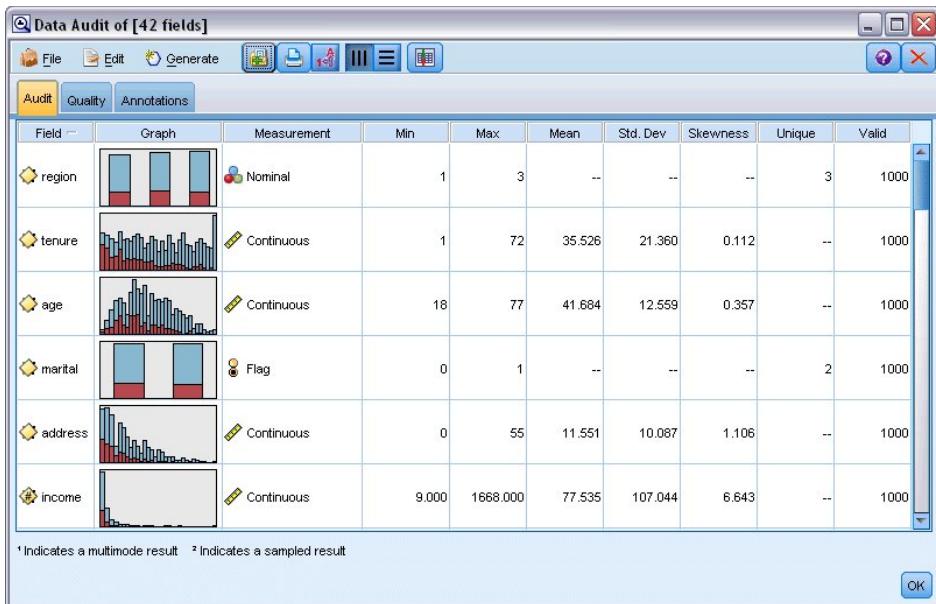


[Next](#)

Browsing Statistics and Charts

The Data Audit browser is displayed, with thumbnail graphs and descriptive statistics for each field.

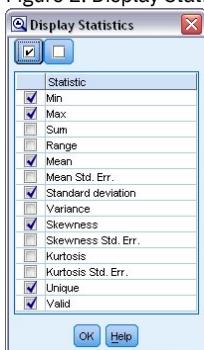
Figure 1. Data Audit browser



Use the toolbar to display field and value labels, and to toggle the alignment of charts from horizontal to vertical (for categorical fields only).

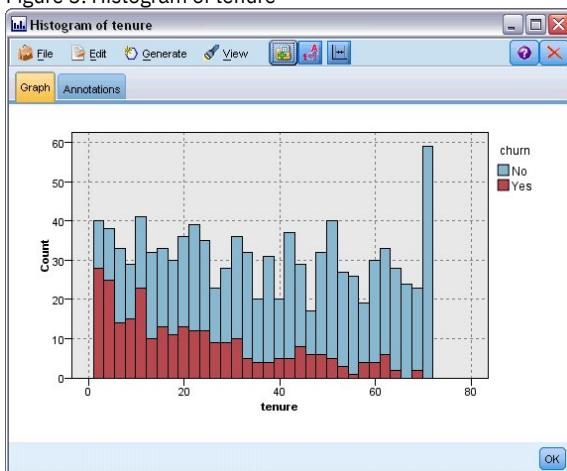
1. You can also use the toolbar or Edit menu to choose the statistics to display.

Figure 2. Display Statistics



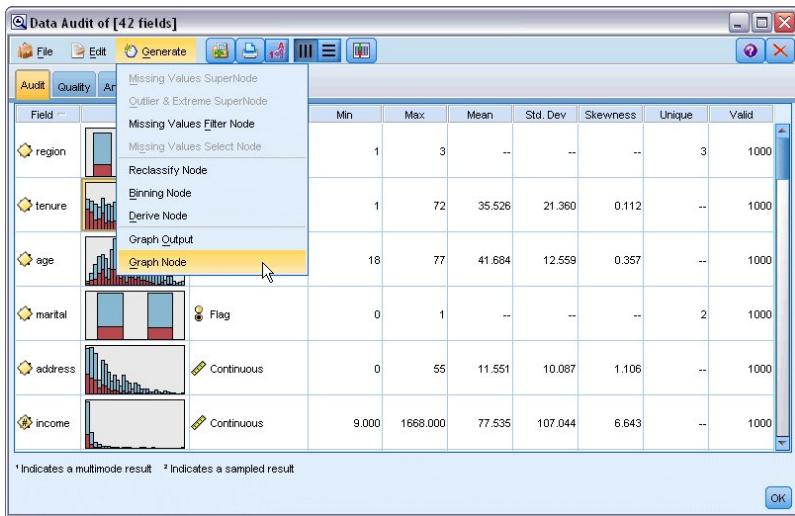
Double-click on any thumbnail graph in the audit report to view a full-sized version of that chart. Because *churn* is the only target field in the stream, it is automatically used as an overlay. You can toggle the display of field and value labels using the graph window toolbar, or click the Edit mode button to further customize the chart.

Figure 3. Histogram of tenure



Alternatively, you can select one or more thumbnails and generate a Graph node for each. The generated nodes are placed on the stream canvas and can be added to the stream to re-create that particular graph.

Figure 4. Generating a Graph node



[Next](#)

Handling Outliers and Missing Values

The Quality tab in the audit report displays information about outliers, extremes, and missing values.

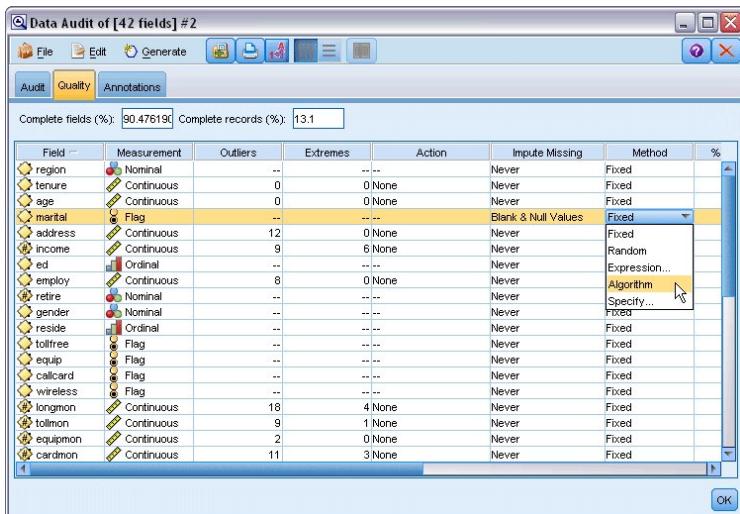
Figure 1. Data Audit browser, Quality tab

The screenshot shows the Data Audit browser window with the Quality tab selected. The top bar shows 'Complete fields (%)' at 90.47619 and 'Complete records (%)' at 13.1. The main table has columns: Field, Measurement, Outliers, Extremes, Action, Impute Missing, and Method. The table lists numerous fields with their respective settings for handling missing values.

Field	Measurement	Outliers	Extremes	Action	Impute Missing	Method
region	Nominal	--	--	Never	Fixed	
tenure	Continuous	0	0 None	Never	Fixed	
age	Continuous	0	0 None	Never	Fixed	
marital	Flag	--	--	Never	Fixed	
address	Continuous	12	0 None	Never	Fixed	
income	Continuous	9	6 None	Never	Fixed	
ed	Ordinal	--	--	Never	Fixed	
employ	Continuous	8	0 None	Never	Fixed	
retire	Nominal	--	--	Never	Fixed	
gender	Nominal	--	--	Never	Fixed	
reside	Ordinal	--	--	Never	Fixed	
tollfree	Flag	--	--	Never	Fixed	
equip	Flag	--	--	Never	Fixed	
callcard	Flag	--	--	Never	Fixed	
wireless	Flag	--	--	Never	Fixed	
longmon	Continuous	18	4 None	Never	Fixed	
tolmon	Continuous	9	1 None	Never	Fixed	
equimon	Continuous	2	0 None	Never	Fixed	
cardmon	Continuous	11	3 None	Never	Fixed	

You can also specify methods for handling these values and generate SuperNodes to automatically apply the transformations. For example you can select one or more fields and choose to impute or replace missing values for these fields using a number of methods, including the C&RT algorithm.

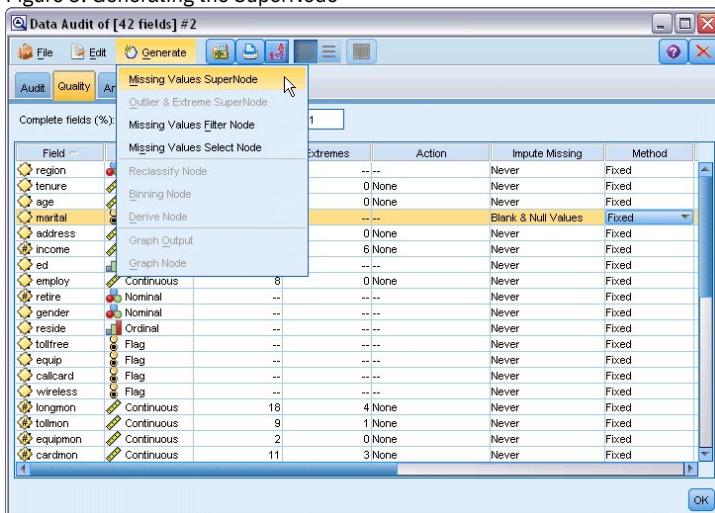
Figure 2. Choosing an impute method



After specifying an impute method for one or more fields, to generate a Missing Values SuperNode, from the menus choose:

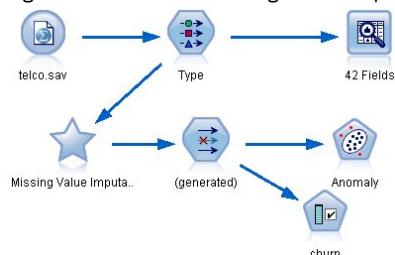
Generate > Missing Values SuperNode

Figure 3. Generating the SuperNode



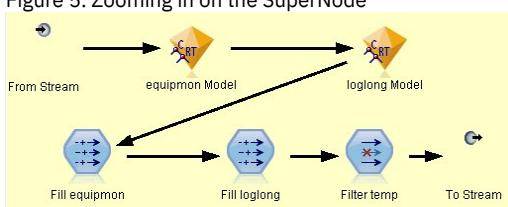
The generated SuperNode is added to the stream canvas, where you can attach it to the stream to apply the transformations.

Figure 4. Stream with Missing Values SuperNode



The SuperNode actually contains a series of nodes that perform the requested transformations. To understand how it works, you can edit the SuperNode and click Zoom In.

Figure 5. Zooming in on the SuperNode



For each field imputed using the algorithm method, for example, there will be a separate C&RT model, along with a Filler node that replaces blanks and nulls with the value predicted by the model. You can add, edit, or remove specific nodes within the SuperNode to further customize the behavior.

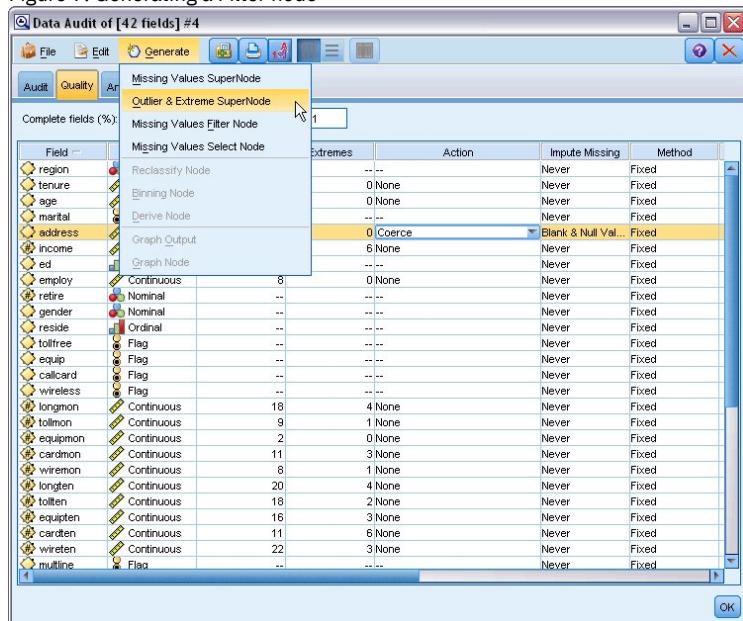
Alternatively, you can generate a Select or Filter node to remove fields or records with missing values. For example, you can filter any fields with a quality percentage below a specified threshold.

Figure 6. Generating a Filter node



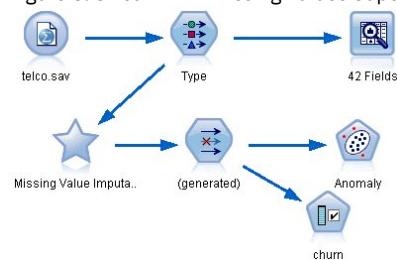
Outliers and extreme values can be handled in a similar manner. Specify the action you want to take for each field—either coerce, discard, or nullify—and generate a SuperNode to apply the transformations.

Figure 7. Generating a Filter node



After completing the audit and adding the generated nodes to the stream, you can proceed with your analysis. Optionally, you may want to further screen your data using Anomaly Detection, Feature Selection, or a number of other methods.

Figure 8. Stream with Missing Values SuperNode



Drug Treatments (Exploratory Graphs/C5.0)

For this section, imagine that you are a medical researcher compiling data for a study. You have collected data about a set of patients, all of whom suffered from the same illness. During their course of treatment, each patient responded to one of five medications. Part of your job is to use data mining to find out which drug might be appropriate for a future patient with the same illness.

This example uses the stream named *druglearn.str*, which references the data file named *DRUG1n*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *druglearn.str* file is in the *streams* directory.

The data fields used in the demo are:

Data field	Description
Age	(Number)
Sex	M or F
BP	Blood pressure: HIGH, NORMAL, or LOW
Cholesterol	Blood cholesterol: NORMAL or HIGH
Na	Blood sodium concentration
K	Blood potassium concentration
Drug	Prescription drug to which a patient responded

[Next](#)

- [Reading in Text Data](#)
- [Adding a Table](#)
- [Creating a Distribution Graph](#)
- [Creating a Scatterplot](#)
- [Creating a Web Graph](#)
- [Deriving a New Field](#)
- [Building a Model](#)
- [Browsing the model](#)
- [Using an Analysis Node](#)

Reading in Text Data

Figure 1. Adding a Variable File node



You can read in delimited text data using a **Variable File node**. You can add a Variable File node from the palettes--either click the Sources tab to find the node or use the Favorites tab, which includes this node by default. Next, double-click the newly placed node to open its dialog box.

Click the button just to the right of the File box marked with an ellipsis (...) to browse to the directory in which IBM® SPSS® Modeler is installed on your system. Open the *Demos* directory and select the file called *DRUG1n*.

Ensuring that Read field names from file is selected, notice the fields and values that have just been loaded into the dialog box.

Figure 2. Variable File dialog box

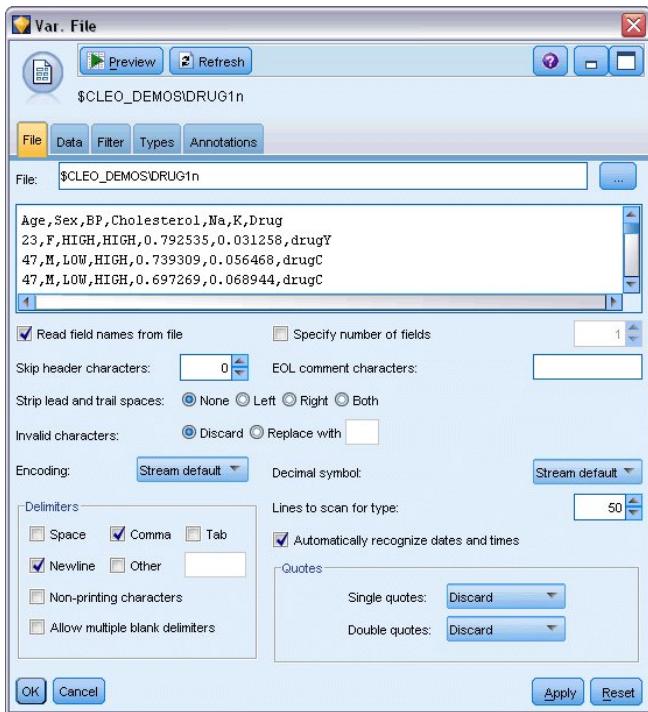


Figure 3. Changing the storage type for a field

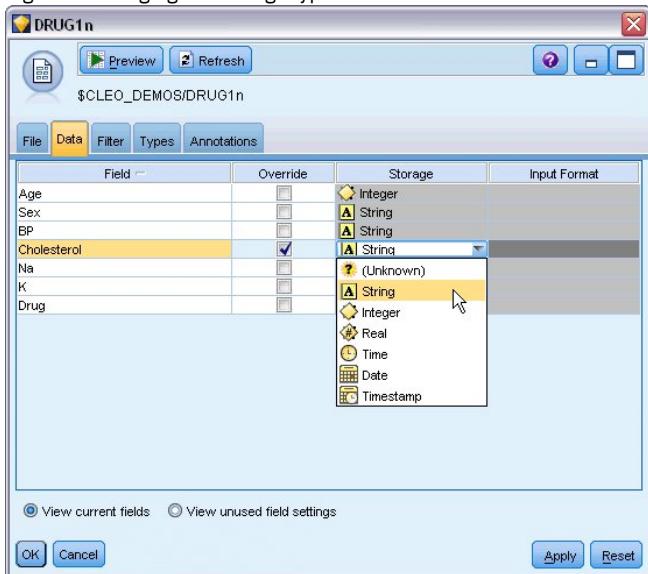


Figure 4. Selecting Value options on the Types tab

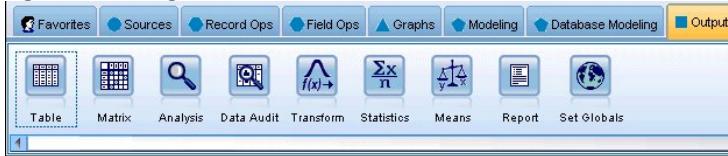


Click the Data tab to override and change **Storage** for a field. Note that storage is different from **Measurement**, that is, the measurement level (or usage type) of the data field. The Types tab helps you learn more about the type of fields in your data. You can also choose Read Values to view the actual values for each field based on the selections that you make from the *Values* column. This process is known as **instantiation**.

[Next](#)

Adding a Table

Figure 1. Selecting the Table node



Now that you have loaded the data file, you may want to glance at the values for some of the records. One way to do this is by building a stream that includes a Table node. To place a Table node in the stream, either double-click the icon in the palette or drag and drop it on to the canvas.

Figure 2. Table node connected to the data source

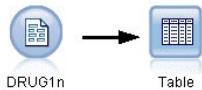


Figure 3. Running a stream from the toolbar

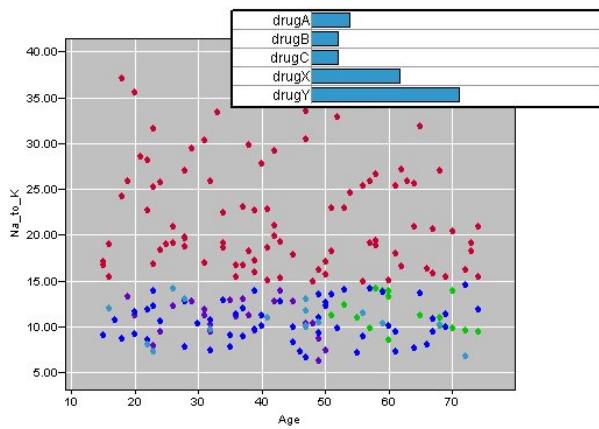
Double-clicking a node from the palette will automatically connect it to the selected node in the stream canvas. Alternatively, if the nodes are not already connected, you can use your middle mouse button to connect the Source node to the Table node. To simulate a middle mouse button, hold down the Alt key while using the mouse. To view the table, click the green arrow button on the toolbar to run the stream, or right-click the Table node and choose Run.

[Next](#)

Creating a Distribution Graph

During data mining, it is often useful to explore the data by creating visual summaries. IBM® SPSS® Modeler offers several different types of graphs to choose from, depending on the kind of data that you want to summarize. For example, to find out what proportion of the patients responded to each drug, use a Distribution node.

Figure 1. Sample of available graphs



Add a Distribution node to the stream and connect it to the Source node, then double-click the node to edit options for display.

Select *Drug* as the target field whose distribution you want to show. Then, click Run from the dialog box.

Figure 2. Selecting drug as the target field

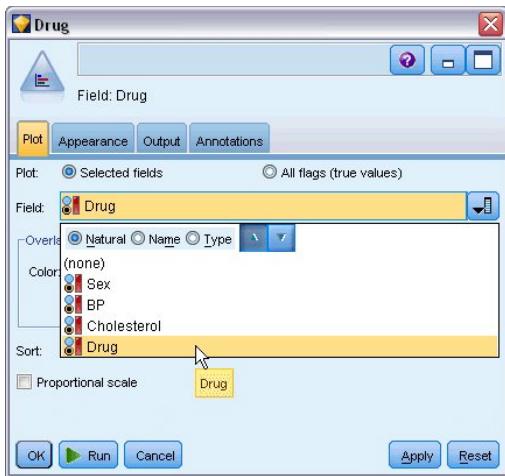
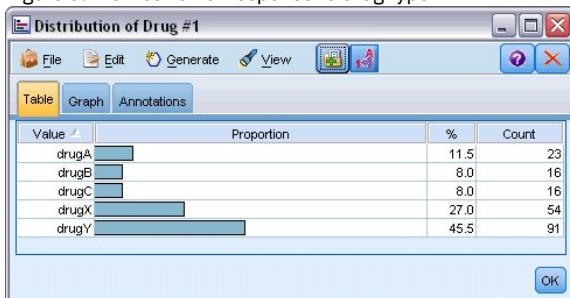


Figure 3. Distribution of response to drug type



The resulting graph helps you see the "shape" of the data. It shows that patients responded to drug Y most often and to drugs B and C least often.

Figure 4. Results of a data audit

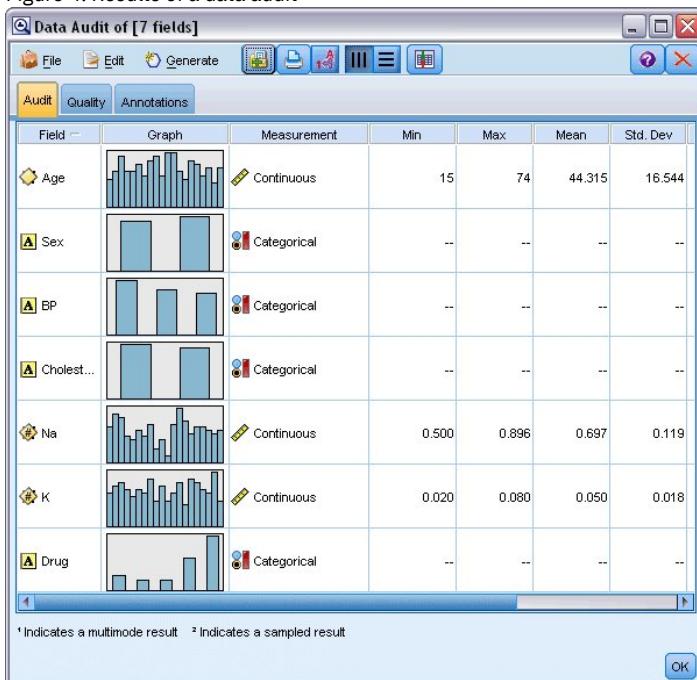
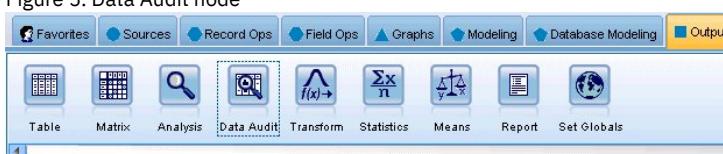
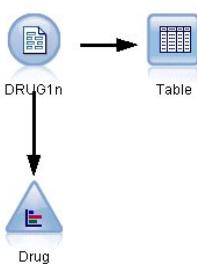


Figure 5. Data Audit node



Alternatively, you can attach and execute a Data Audit node for a quick glance at distributions and histograms for all fields at once. The Data Audit node is available on the Output tab.

Creating a Scatterplot



Now let's take a look at what factors might influence *Drug*, the target variable. As a researcher, you know that the concentrations of sodium and potassium in the blood are important factors. Since these are both numeric values, you can create a scatterplot of sodium versus potassium, using the drug categories as a color overlay.

Place a Plot node in the workspace and connect it to the Source node, and double-click to edit the node.

Figure 1. Stream with plot node

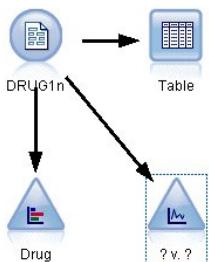
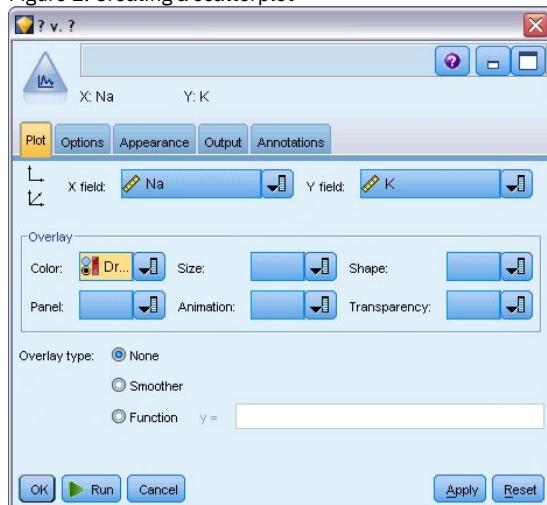


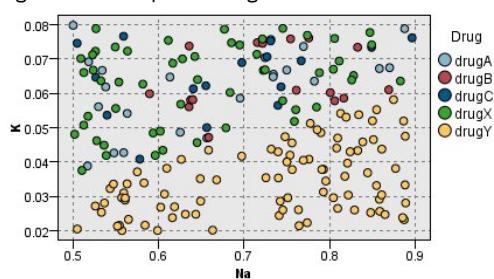
Figure 2. Creating a scatterplot



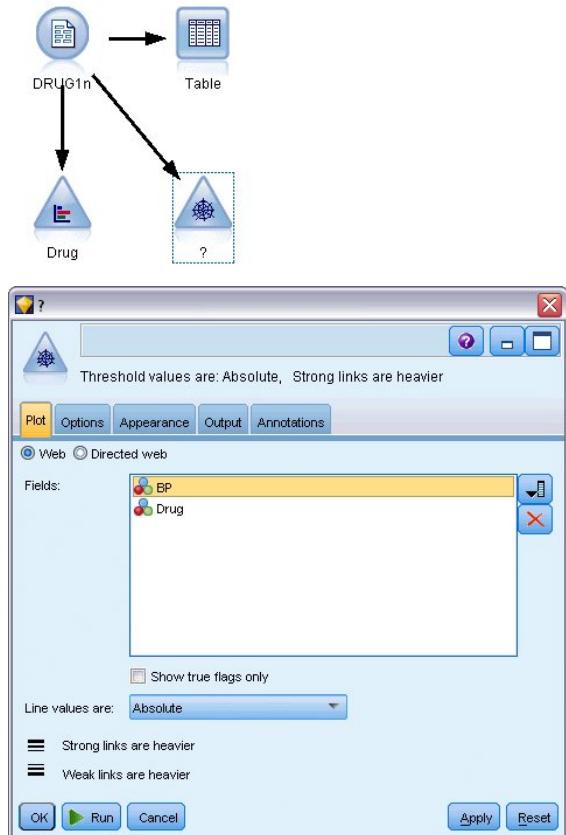
On the Plot tab, select *Na* as the X field, *K* as the Y field, and *Drug* as the overlay field. Then, click Run.

The plot clearly shows a threshold above which the correct drug is always drug Y and below which the correct drug is never drug Y. This threshold is a ratio--the ratio of sodium (*Na*) to potassium (*K*).

Figure 3. Scatterplot of drug distribution

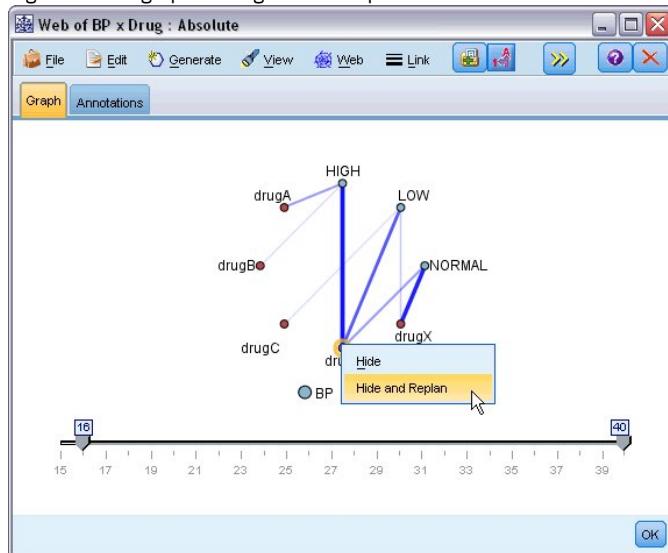


Creating a Web Graph



Since many of the data fields are categorical, you can also try plotting a web graph, which maps associations between different categories. Start by connecting a Web node to the Source node in your workspace. In the Web node dialog box, select *BP* (for blood pressure) and *Drug*. Then, click Run.

Figure 1. Web graph of drugs vs. blood pressure

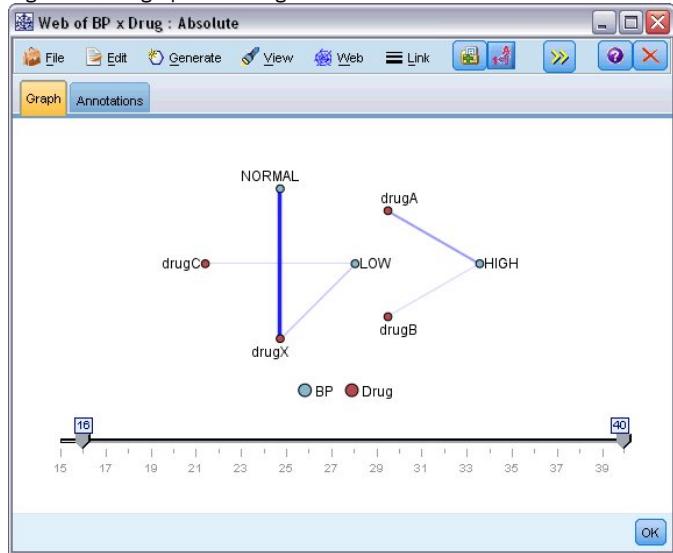


From the plot, it appears that drug Y is associated with all three levels of blood pressure. This is no surprise--you have already determined the situation in which drug Y is best. To focus on the other drugs, you can hide drug Y. On the View menu, choose Edit Mode, then right-click over the drug Y point and choose Hide and Replan.

In the simplified plot, drug Y and all of its links are hidden. Now, you can clearly see that only drugs A and B are associated with high blood pressure. Only drugs C and X are associated with low blood pressure. And normal blood pressure is associated only with drug X. At this point,

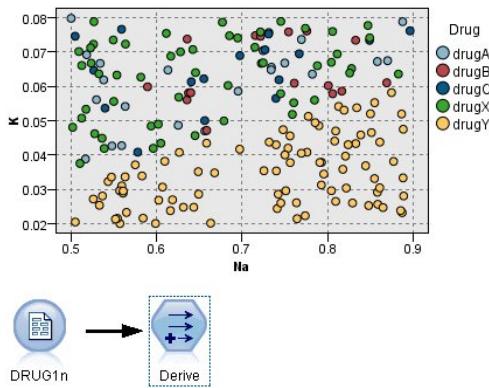
though, you still don't know how to choose between drugs *A* and *B* or between drugs *C* and *X*, for a given patient. This is where modeling can help.

Figure 2. Web graph with drug Y and its links hidden



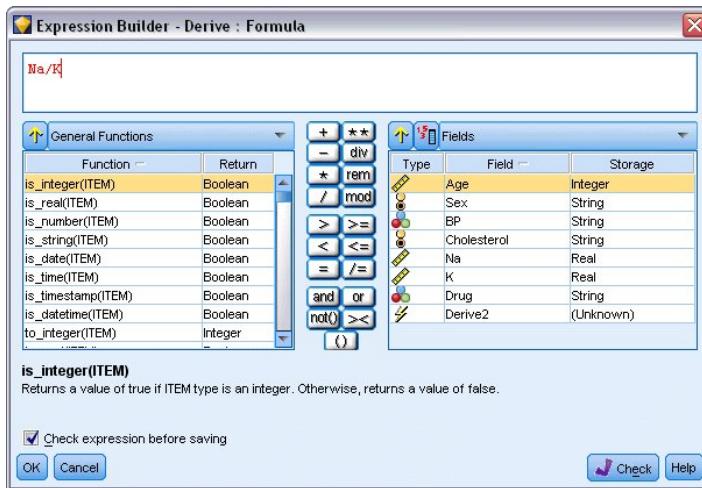
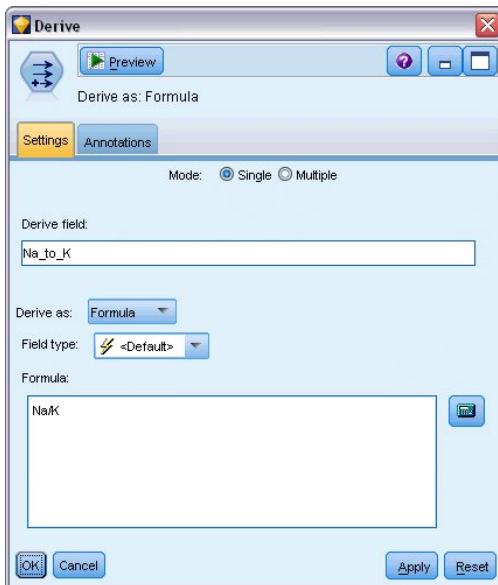
[Next](#)

Deriving a New Field



Since the ratio of sodium to potassium seems to predict when to use drug *Y*, you can derive a field that contains the value of this ratio for each record. This field might be useful later when you build a model to predict when to use each of the five drugs. To simplify the stream layout, start by deleting all the nodes except the DRUG1n source node. Attach a Derive node (Field Ops tab) to DRUG1n, then double-click the Derive node to edit it.

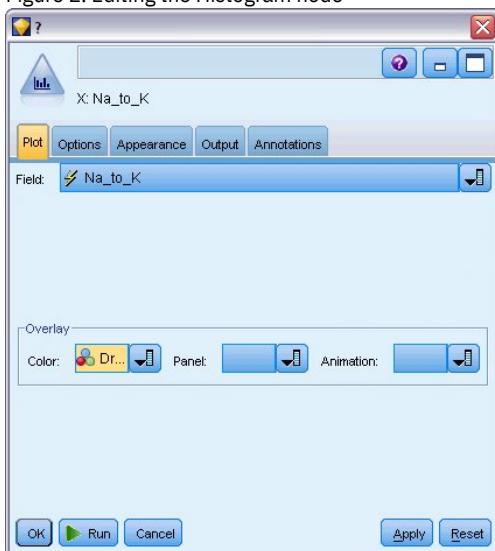
Figure 1. Editing the Derive node



Name the new field *Na_to_K*. Since you obtain the new field by dividing the sodium value by the potassium value, enter *Na/K* for the formula. You can also create a formula by clicking the icon just to the right of the field. This opens the Expression Builder, a way to interactively create expressions using built-in lists of functions, operands, and fields and their values.

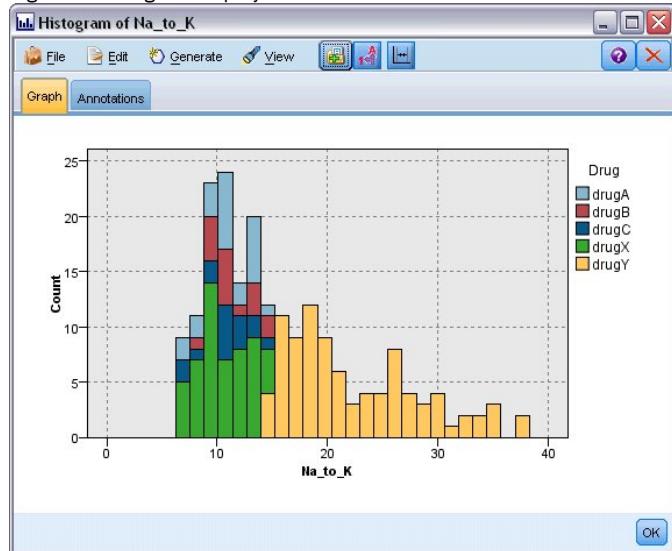
You can check the distribution of your new field by attaching a Histogram node to the Derive node. In the Histogram node dialog box, specify *Na_to_K* as the field to be plotted and *Drug* as the overlay field.

Figure 2. Editing the Histogram node



When you run the stream, you get the graph shown here. Based on the display, you can conclude that when the *Na_to_K* value is about 15 or above, drug Y is the drug of choice.

Figure 3. Histogram display



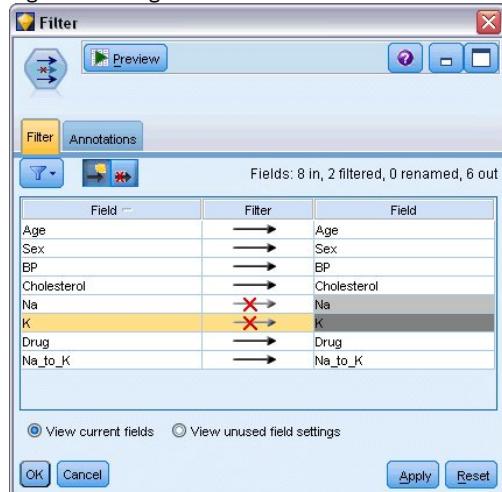
[Next](#)

Building a Model

By exploring and manipulating the data, you have been able to form some hypotheses. The ratio of sodium to potassium in the blood seems to affect the choice of drug, as does blood pressure. But you cannot fully explain all of the relationships yet. This is where modeling will likely provide some answers. In this case, you will use try to fit the data using a rule-building model, C5.0.

Since you are using a derived field, *Na_to_K*, you can filter out the original fields, *Na* and *K*, so that they are not used twice in the modeling algorithm. You can do this using a Filter node.

Figure 1. Editing the Filter node

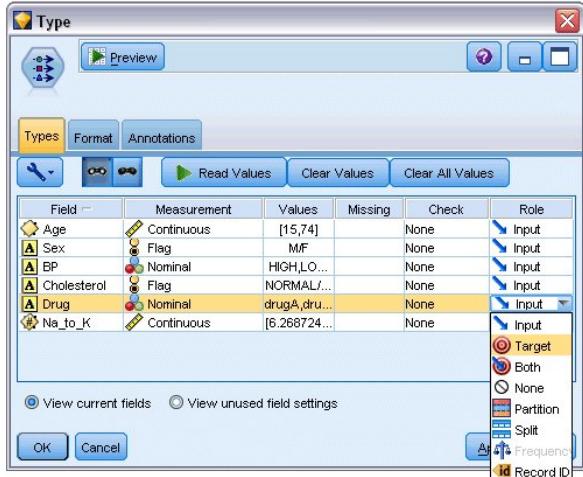


On the Filter tab, click the arrows next to *Na* and *K*. Red Xs appear over the arrows to indicate that the fields are now filtered out.

Next, attach a Type node connected to the Filter node. The Type node allows you to indicate the types of fields that you are using and how they are used to predict the outcomes.

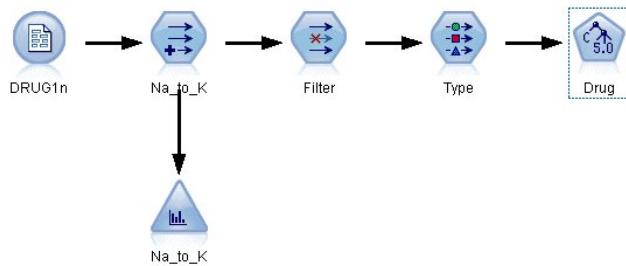
On the Types tab, set the role for the *Drug* field to Target, indicating that *Drug* is the field you want to predict. Leave the role for the other fields set to Input so they will be used as predictors.

Figure 2. Editing the Type node



To estimate the model, place a C5.0 node in the workspace and attach it to the end of the stream as shown. Then click the green Run toolbar button to run the stream.

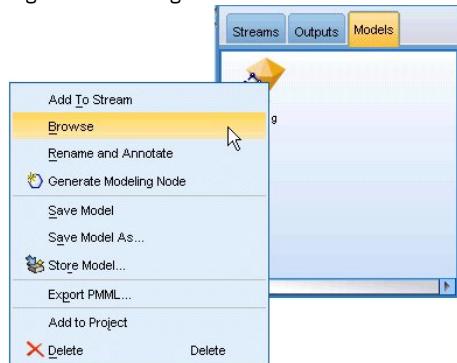
Figure 3. Adding a C5.0 node



[Next](#)

Browsing the model

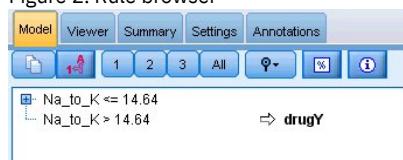
Figure 1. Browsing the model



When the C5.0 node is executed, the model nugget is added to the stream, and also to the Models palette in the upper-right corner of the window. To browse the model, right-click either of the icons and choose Edit or Browse from the context menu.

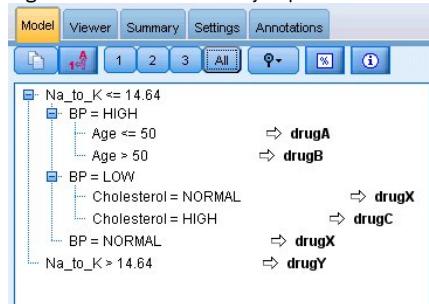
The Rule browser displays the set of rules generated by the C5.0 node in a decision tree format. Initially, the tree is collapsed. To expand it, click the All button to show all levels.

Figure 2. Rule browser



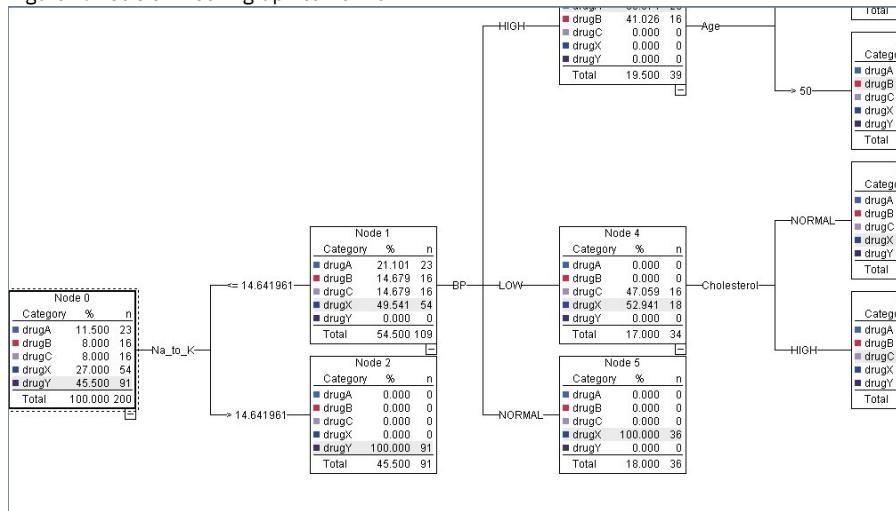
Now you can see the missing pieces of the puzzle. For people with an Na-to-K ratio less than 14.64 and high blood pressure, age determines the choice of drug. For people with low blood pressure, cholesterol level seems to be the best predictor.

Figure 3. Rule browser fully expanded



The same decision tree can be viewed in a more sophisticated graphical format by clicking the Viewer tab. Here, you can see more easily the number of cases for each blood pressure category, as well as the percentage of cases.

Figure 4. Decision tree in graphical format

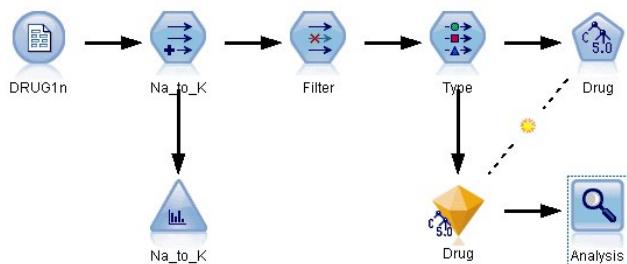


[Next](#)

Using an Analysis Node

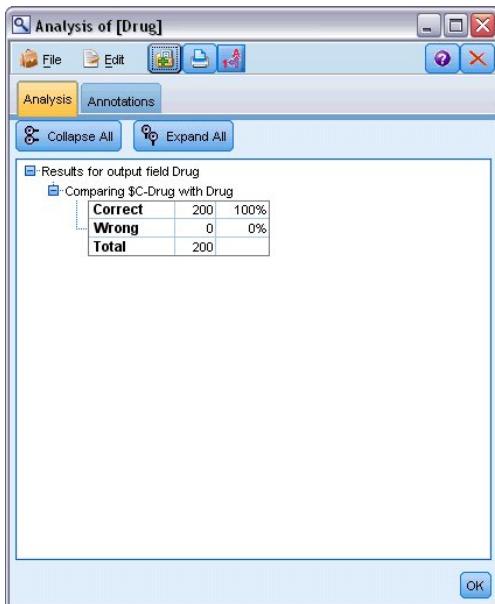
You can assess the accuracy of the model using an analysis node. Attach an Analysis node (from the Output node palette) to the model nugget, open the Analysis node and click Run.

Figure 1. Adding an Analysis node



The Analysis node output shows that with this artificial dataset, the model correctly predicted the choice of drug for every record in the dataset. With a real dataset you are unlikely to see 100% accuracy, but you can use the Analysis node to help determine whether the model is acceptably accurate for your particular application.

Figure 2. Analysis node output



Screening Predictors (Feature Selection)

The Feature Selection node helps you to identify the fields that are most important in predicting a certain outcome. From a set of hundreds or even thousands of predictors, the Feature Selection node screens, ranks, and selects the predictors that may be most important. Ultimately, you may end up with a quicker, more efficient model—one that uses fewer predictors, executes more quickly, and may be easier to understand.

The data used in this example represent a data warehouse for a hypothetical telephone company and contain information about responses to a special promotion by 5,000 of the company's customers. The data include a large number of fields containing customers' age, employment, income, and telephone usage statistics. Three "target" fields show whether or not the customer responded to each of three offers. The company wants to use this data to help predict which customers are most likely to respond to similar offers in the future.

This example uses the stream named *featureselection.str*, which references the data file named *customer_dbase.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *featureselection.str* file is in the *streams* directory.

This example focuses on only one of the offers as a target. It uses the CHAID tree-building node to develop a model to describe which customers are most likely to respond to the promotion. It contrasts two approaches:

- Without feature selection. All predictor fields in the dataset are used as inputs to the CHAID tree.
- With feature selection. The Feature Selection node is used to select the top 10 predictors. These are then input into the CHAID tree.

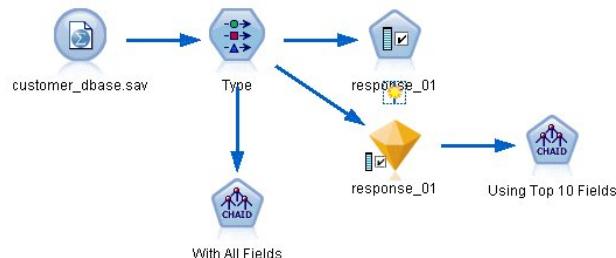
By comparing the two resulting tree models, we can see how feature selection produces effective results.

[Next](#)

- [Building the Stream](#)
- [Building the Models](#)
- [Comparing the Results](#)
- [Summary](#)

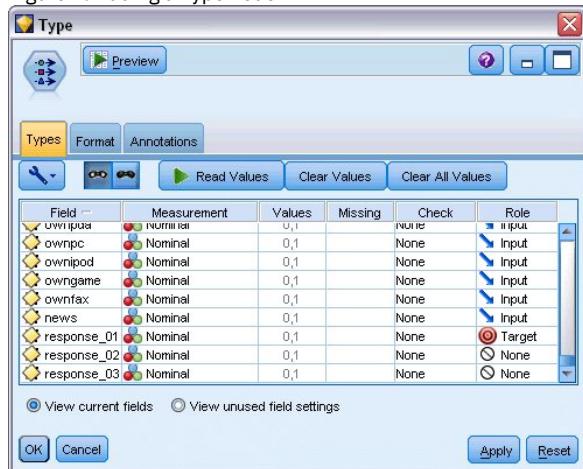
Building the Stream

Figure 1. Feature Selection example stream



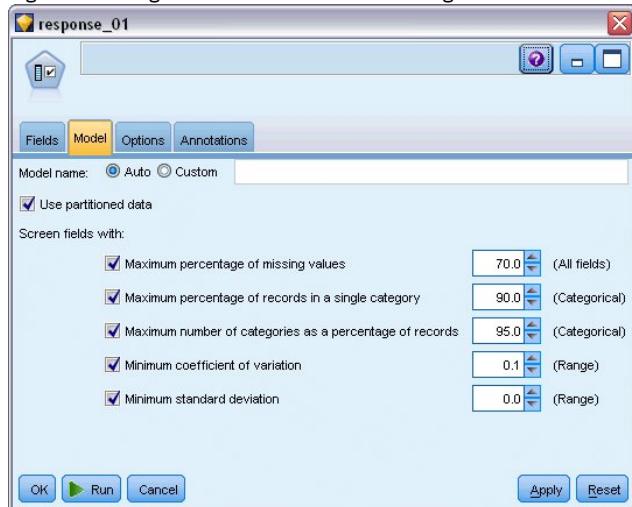
1. Place a Statistics File source node onto a blank stream canvas. Point this node to the example data file *customer_dbase.sav*, available in the *Demos* directory under your IBM® SPSS® Modeler installation. (Alternatively, open the example stream file *featureselection.str* in the *streams* directory.)
2. Add a Type node. On the Types tab, scroll down to the bottom and change the role for *response_01* to *Target*. Change the role to *None* for the other response fields (*response_02* and *response_03*) as well as for the customer ID (*custid*) at the top of the list. Leave the role set to *Input* for all other fields, and click the Read Values button, then click OK.

Figure 2. Adding a Type node



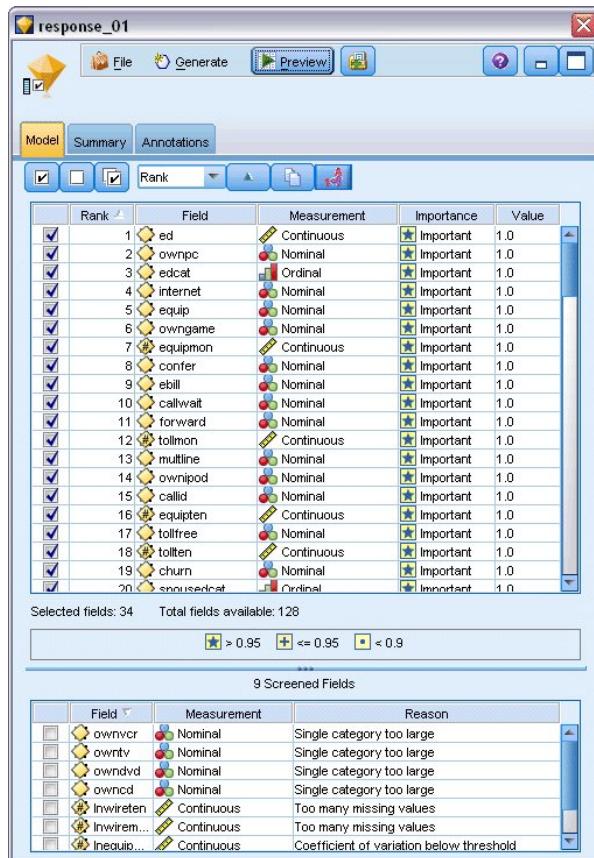
3. Add a Feature Selection modeling node to the stream. On this node, you can specify the rules and criteria for screening, or disqualifying, fields.

Figure 3. Adding the Feature Selection modeling node



4. Run the stream to create the Feature Selection model nugget.
5. Right-click the model nugget on the stream or in the Models palette and choose Edit or Browse to look at the results.

Figure 4. Model tab in Feature Selection model nugget

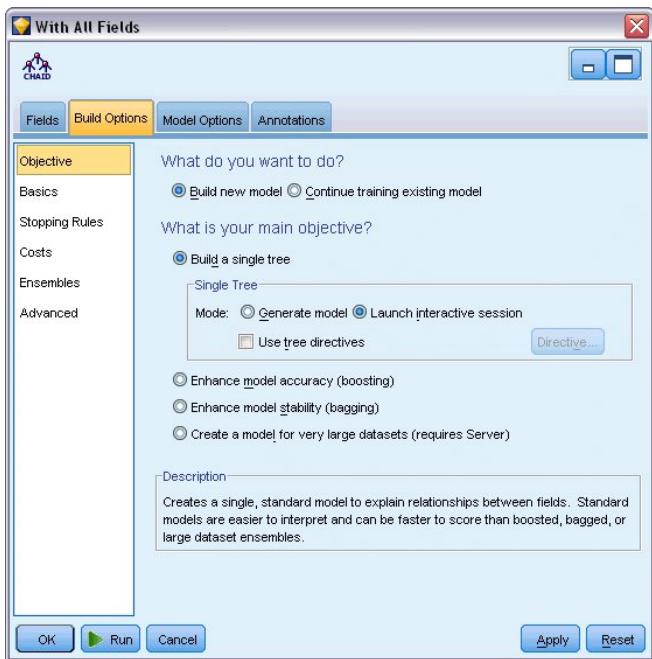


The top panel shows the fields found to be useful in the prediction. These are ranked based on importance. The bottom panel shows which fields were screened from the analysis and why. By examining the fields in the top panel, you can decide which ones to use in subsequent modeling sessions.

6. Now we can select the fields to use downstream. Although 34 fields were originally identified as important, we want to reduce the set of predictors even further.
7. Select only the top 10 predictors using the check marks in the first column to deselect the unwanted predictors. (Click the check mark in row 11, hold down the Shift key and click the check mark in row 34.) Close the model nugget.
8. To compare results without feature selection, you must add two CHAID modeling nodes to the stream: one that uses feature selection and one that does not.
9. Connect one CHAID node to the Type node, and the other one to the Feature Selection model nugget.
10. Open each CHAID node, select the Build Options tab and ensure that the options Build new model, Build a single tree and Launch interactive session are selected in the Objectives pane.

On the Basics pane, make sure that Maximum Tree Depth is set to 5.

Figure 5. Objectives settings for CHAID modeling node for all predictor fields

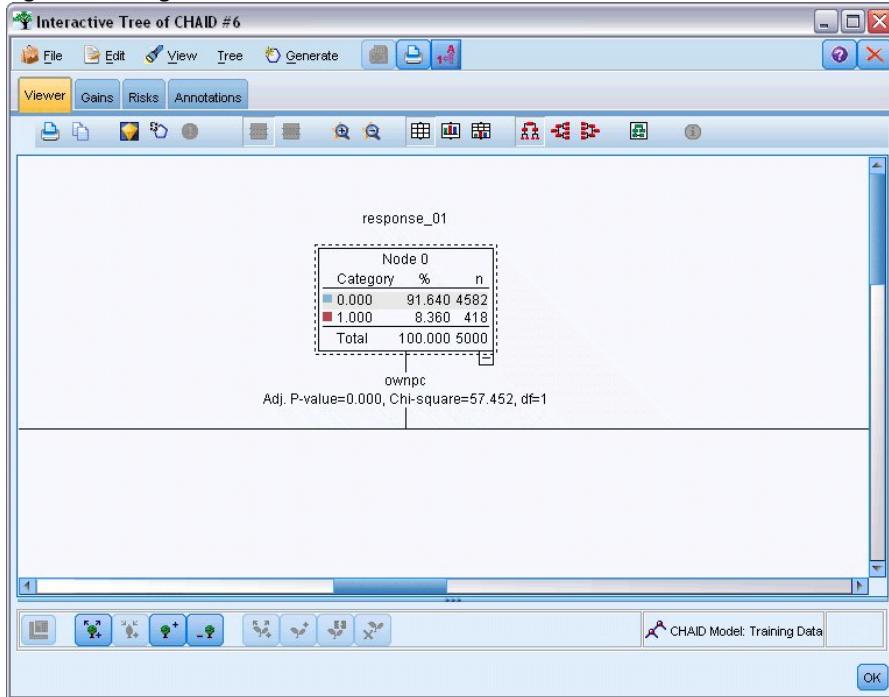


[Next](#)

Building the Models

1. Execute the CHAID node that uses all of the predictors in the dataset (the one connected to the Type node). As it runs, notice how long it takes to execute. The results window displays a table.
2. From the menus, choose Tree > Grow Tree to grow and display the expanded tree.

Figure 1. Growing the tree in the Tree Builder



3. Now do the same for the other CHAID node, which uses only 10 predictors. Again, grow the tree when the Tree Builder opens.

The second model should have executed faster than the first one. Because this dataset is fairly small, the difference in execution times is probably a few seconds; but for larger real-world datasets, the difference may be very noticeable—minutes or even hours. Using feature selection may speed up your processing times dramatically.

The second tree also contains fewer tree nodes than the first. It is easier to comprehend. But before you decide to use it, you need to find out whether it is effective and how it compares to the model that uses all predictors.

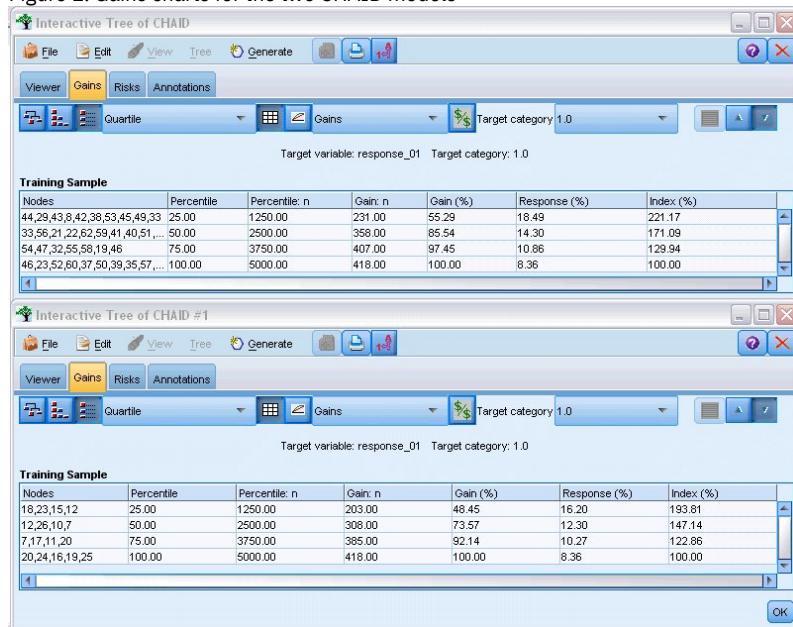
[Next](#)

Comparing the Results

To compare the two results, we need a measure of effectiveness. For this, we will use the Gains tab in the Tree Builder. We will look at **lift**, which measures how much more likely the records in a node are to fall under the target category when compared to all records in the dataset. For example, a lift value of 148% indicates that records in the node are 1.48 times more likely to fall under the target category than all records in the dataset. Lift is indicated in the *Index* column on the Gains tab.

1. In the Tree Builder for the full set of predictors, click the Gains tab. Change the target category to 1.0. Change the display to quartiles by first clicking the Quantiles toolbar button. Then select Quartile from the drop-down list to the right of this button.
2. Repeat this procedure in the Tree Builder for the set of 10 predictors so that you have two similar Gains tables to compare, as shown in the following figures.

Figure 1. Gains charts for the two CHAID models



Each Gains table groups the terminal nodes for its tree into quartiles. To compare the effectiveness of the two models, look at the lift (*Index* value) for the top quartile in each table.

When all predictors are included, the model shows a lift of 221%. That is, cases with the characteristics in these nodes are 2.2 times more likely to respond to the target promotion. To see what those characteristics are, click to select the top row. Then switch to the Viewer tab, where the corresponding nodes are now outlined in black. Follow the tree down to each highlighted terminal node to see how the predictors were split. The top quartile alone includes 10 nodes. When translated into real-world scoring models, 10 different customer profiles can be difficult to manage.

With only the top 10 predictors (as identified by feature selection) included, the lift is nearly 194%. Although this model is not quite as good as the model that uses all predictors, it is certainly useful. Here, the top quartile includes only four nodes, so it is simpler. Therefore, we can determine that the feature selection model is preferable to the one with all predictors.

[Next](#)

Summary

Let's review the advantages of feature selection. Using fewer predictors is less expensive. It means that you have less data to collect, process, and feed into your models. Computing time is improved. In this example, even with the extra feature selection step, model building was noticeably faster with the smaller set of predictors. With a larger real-world dataset, the time savings should be greatly amplified.

Using fewer predictors results in simpler scoring. As the example shows, you might identify only four profiles of customers who are likely to respond to the promotion. Note that with larger numbers of predictors, you run the risk of overfitting your model. The simpler model may generalize better to other datasets (although you would need to test this to be sure).

You could have used a tree-building algorithm to do the feature selection work, allowing the tree to identify the most important predictors for you. In fact, the CHAID algorithm is often used for this purpose, and it is even possible to grow the tree level-by-level to control its depth and complexity. However, the Feature Selection node is faster and easier to use. It ranks all of the predictors in one fast step, allowing you to identify the most important fields quickly. It also allows you to vary the number of predictors to include. You could easily run this example again using the top 15 or 20 predictors instead of 10, comparing the results to determine the optimal model.

Reducing Input Data String Length (Reclassify Node)

- #### **• Reducing Input Data String Length (Reclassify)**

Reducing Input Data String Length (Reclassify)

For binomial logistic regression, and auto classifier models that include a binomial logistic regression model, string fields are limited to a maximum of eight characters. Where strings are more than eight characters, they can be recoded using a Reclassify node.

This example uses the stream named *reclassify_strings.str*, which references the data file named *drug_long_name*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *reclassify_strings.str* file is in the *streams* directory.

This example focuses on a small part of a stream to show the sort of errors that may be generated with overlong strings and explains how to use the Reclassify node to change the string details to an acceptable length. Although the example uses a binomial Logistic Regression node, it is equally applicable when using the Auto Classifier node to generate a binomial Logistic Regression model.

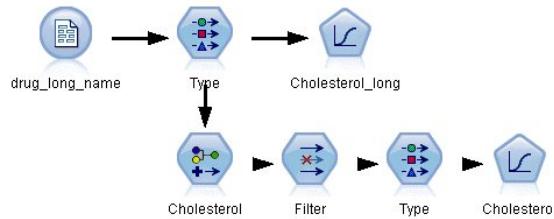
Next

- Reclassifying the Data

Reclassifying the Data

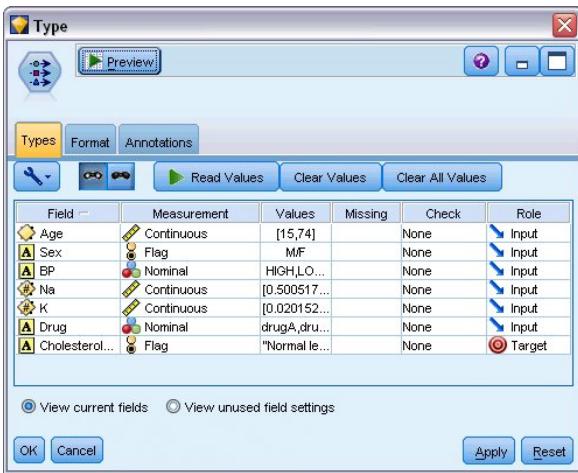
1. Using a Variable File source node, connect to the dataset *drug_long_name* in the *Demos* folder.

Figure 1. Sample stream showing string reclassification for binomial logistic regression



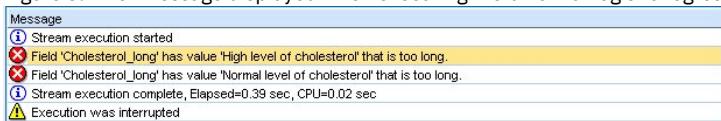
2. Add a Type node to the Source node and select Cholesterol_long as the target.
 3. Add a Logistic Regression node to the Type node.
 4. In the Logistic Regression node, click the Model tab and select the Binomial procedure.

Figure 2. Long string details in the "Cholesterol_long" field



5. When you execute the Logistic Regression node in *reclassify_strings.str*, an error message is displayed warning that the Cholesterol_long string values are too long.
If you encounter this type of error message, follow the procedure explained in the rest of this example to modify your data.

Figure 3. Error message displayed when executing the binomial logistic regression node



6. Add a Reclassify node to the Type node.
7. In the Reclassify field, select Cholesterol_long.
8. Type Cholesterol as the new field name.
9. Click the Get button to add the Cholesterol_long values to the original value column.
10. In the new value column, type High next to the original value of High level of cholesterol and Normal next to the original value of Normal level of cholesterol.

Figure 4. Reclassifying the long strings



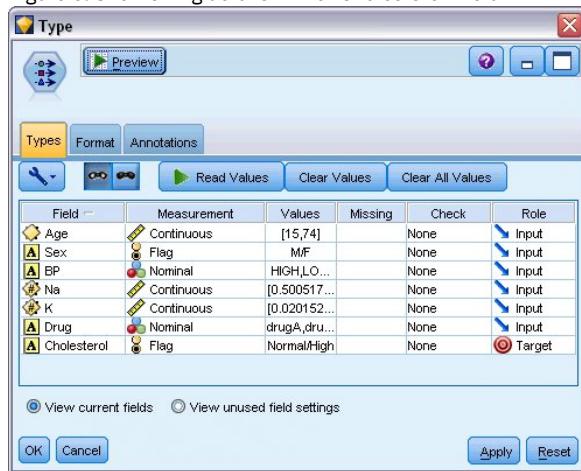
11. Add a Filter node to the Reclassify node.
12. In the Filter column, click to remove Cholesterol_long.

Figure 5. Filtering the "Cholesterol_long" field from the data



13. Add a Type node to the Filter node and select Cholesterol as the target.

Figure 6. Short string details in the "Cholesterol" field

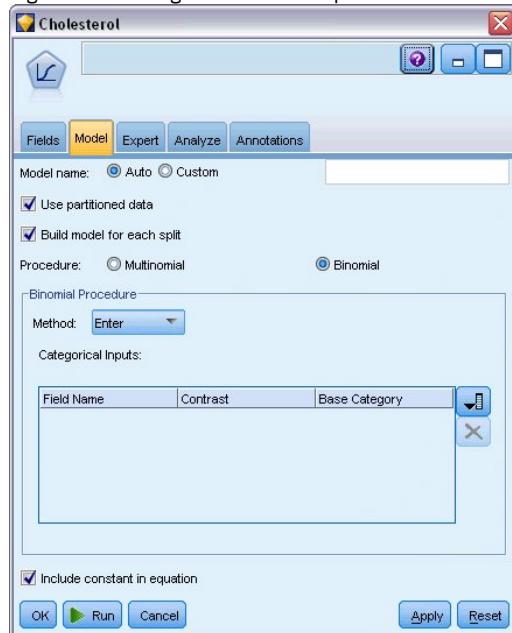


14. Add a Logistic Node to the Type node.

15. In the Logistic node, click the Model tab and select the Binomial procedure.

16. You can now execute the Binomial Logistic node and generate a model without displaying an error message.

Figure 7. Choosing Binomial as the procedure



This example only shows part of a stream. If you require further information about the types of streams in which you may need to reclassify long strings, the following examples are available:

- Auto Classifier node. See the topic [Modeling Customer Response \(Auto Classifier\)](#) for more information.
- Binomial Logistic Regression node. See the topic [Telecommunications Churn \(Binomial Logistic Regression\)](#) for more information.

More information on how to use IBM® SPSS® Modeler, such as a user's guide, node reference, and algorithms guide, are available from the *|Documentation* directory of the installation disk.

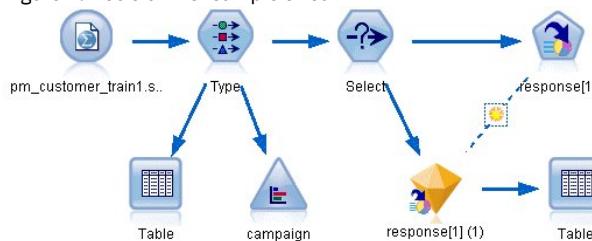
Modeling Customer Response (Decision List)

The Decision List algorithm generates rules that indicate a higher or lower likelihood of a given binary (yes or no) outcome. Decision List models are widely used in customer relationship management, such as call center or marketing applications.

This example is based on a fictional company that wants to achieve more profitable results in future marketing campaigns by matching the right offer to each customer. Specifically, the example uses a Decision List model to identify the characteristics of customers who are most likely to respond favorably, based on previous promotions, and to generate a mailing list based on the results.

Decision List models are particularly well suited to interactive modeling, allowing you to adjust parameters in the model and immediately see the results. For a different approach that allows you to automatically create a number of different models and rank the results, the Auto Classifier node can be used instead.

Figure 1. Decision List sample stream



This example uses the stream *pm_decisionlist.str*, which references the data file *pm_customer_train1.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *pm_decisionlist.str* file is in the *streams* directory.

[Next](#)

- [Historical Data](#)
- [Building the Stream](#)
- [Creating the model](#)
- [Calculating custom measures using Excel](#)
- [Saving the Results](#)

Historical Data

The file *pm_customer_train1.sav* has historical data tracking the offers made to specific customers in past campaigns, as indicated by the value of the *campaign* field. The largest number of records fall under the *Premium account* campaign.

Figure 1. Data about previous promotions

Table (31 fields, 21,927 records)

The screenshot shows a window titled "Table (31 fields, 21,927 records)". The toolbar includes icons for File, Edit, Generate, and various data manipulation tools. Below the toolbar is a tab bar with "Table" and "Annotations". The main area is a grid showing data from a CSV file. The columns include "customer_id", "campaign", "response", "response_date", "purchase", "purchase_date", "product_id", and others. Row 1 shows "customer_id" 7 with "campaign" "Premium account" and "response" 0. Row 2 shows "customer_id" 13 with "campaign" "Premium account" and "response" 0. Row 3 shows "customer_id" 15 with "campaign" "Premium account" and "response" 0. Row 4 shows "customer_id" 16 with "campaign" "Premium account" and "response" 1. Row 5 shows "customer_id" 23 with "campaign" "Premium account" and "response" 0. Row 6 shows "customer_id" 24 with "campaign" "Premium account" and "response" 0. Row 7 shows "customer_id" 30 with "campaign" "Gold card" and "response" 0. Row 8 shows "customer_id" 30 with "campaign" "Gold card" and "response" 0. Row 9 shows "customer_id" 33 with "campaign" "Premium account" and "response" 0. Row 10 shows "customer_id" 42 with "campaign" "Gold card" and "response" 0. Row 11 shows "customer_id" 42 with "campaign" "Premium account" and "response" 0. Row 12 shows "customer_id" 52 with "campaign" "Premium account" and "response" 0. Row 13 shows "customer_id" 57 with "campaign" "Premium account" and "response" 0. Row 14 shows "customer_id" 63 with "campaign" "Premium account" and "response" 1. Row 15 shows "customer_id" 74 with "campaign" "Premium account" and "response" 0. Row 16 shows "customer_id" 74 with "campaign" "Gold card" and "response" 0. Row 17 shows "customer_id" 75 with "campaign" "Premium account" and "response" 0. Row 18 shows "customer_id" 82 with "campaign" "Premium account" and "response" 0. Row 19 shows "customer_id" 89 with "campaign" "Gold card" and "response" 0. Row 20 shows "customer_id" 89 with "campaign" "Premium account" and "response" 0.

The values of the *campaign* field are actually coded as integers in the data, with labels defined in the Type node (for example, 2 = *Premium account*). You can toggle display of value labels in the table using the toolbar.

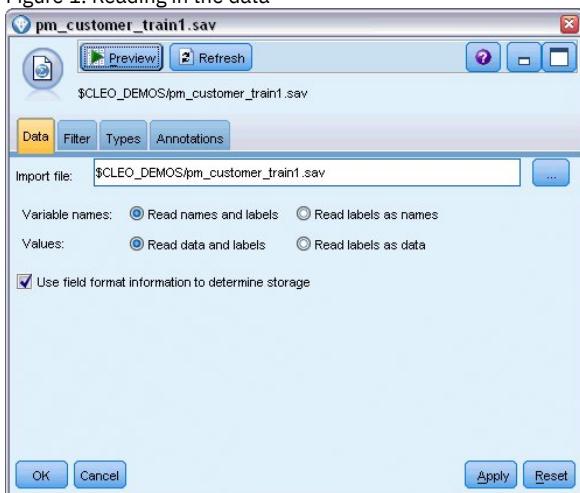
The file also includes a number of fields containing demographic and financial information about each customer that can be used to build or "train" a model that predicts response rates for different groups based on specific characteristics.

[Next](#)

Building the Stream

- Add a Statistics File node pointing to *pm_customer_train1.sav*, located in the *Demos* folder of your IBM® SPSS® Modeler installation. (You can specify **\$CLEO_DEMOS/** in the file path as a shortcut to reference this folder.)

Figure 1. Reading in the data



- Add a Type node, and select *response* as the target field (Role = Target). Set the measurement level for this field to Flag.

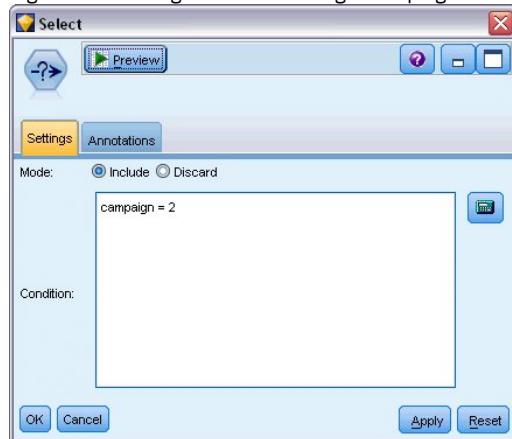
Figure 2. Setting the measurement level and role



3. Set the role to None for the following fields: *customer_id*, *campaign*, *response_date*, *purchase*, *purchase_date*, *product_id*, *Rowid*, and *X_random*. These fields all have uses in the data but will not be used in building the actual model.
4. Click the Read Values button in the Type node to make sure that values are instantiated.

Although the data includes information about four different campaigns, you will focus the analysis on one campaign at a time. Since the largest number of records fall under the Premium campaign (coded *campaign* = 2 in the data), you can use a Select node to include only these records in the stream.

Figure 3. Selecting records for a single campaign

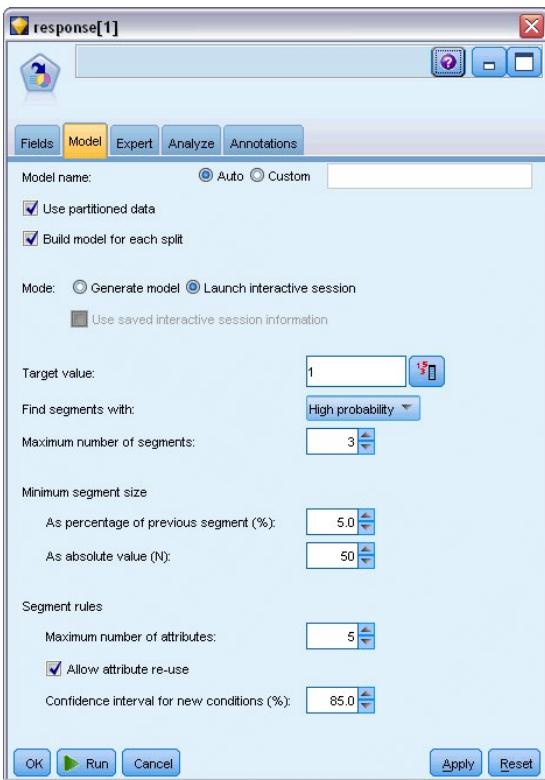


[Next](#)

Creating the model

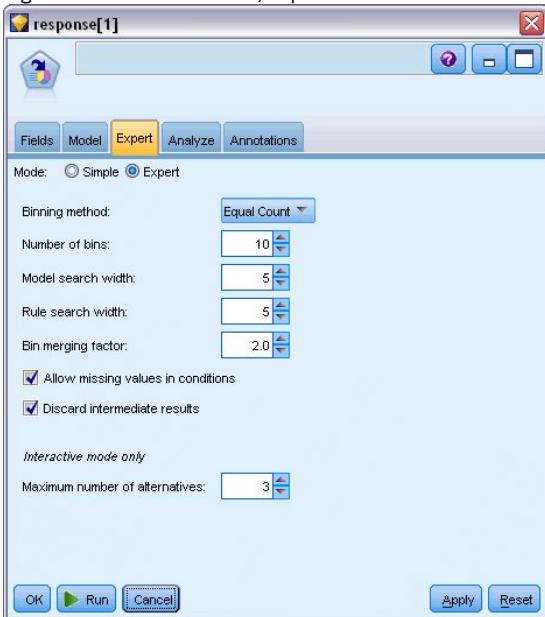
1. Attach a Decision List node to the stream. On the Model tab, set the Target value to 1 to indicate the outcome you want to search for. In this case, you are looking for customers who responded Yes to a previous offer.

Figure 1. Decision List node, Model tab



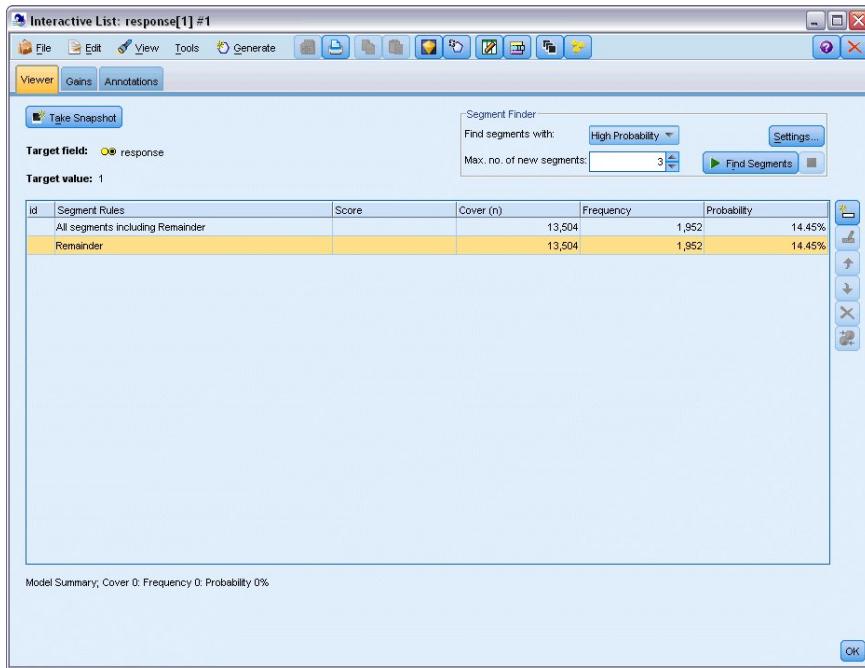
2. Select Launch interactive session.
3. To keep the model simple for purposes of this example, set the maximum number of segments to 3.
4. Change the confidence interval for new conditions to 85%.
5. On the Expert tab, set the Mode to Expert.

Figure 2. Decision List node, Expert tab



6. Increase the Maximum number of alternatives to 3. This option works in conjunction with the Launch interactive session setting that you selected on the Model tab.
7. Click Run to display the Interactive List viewer.

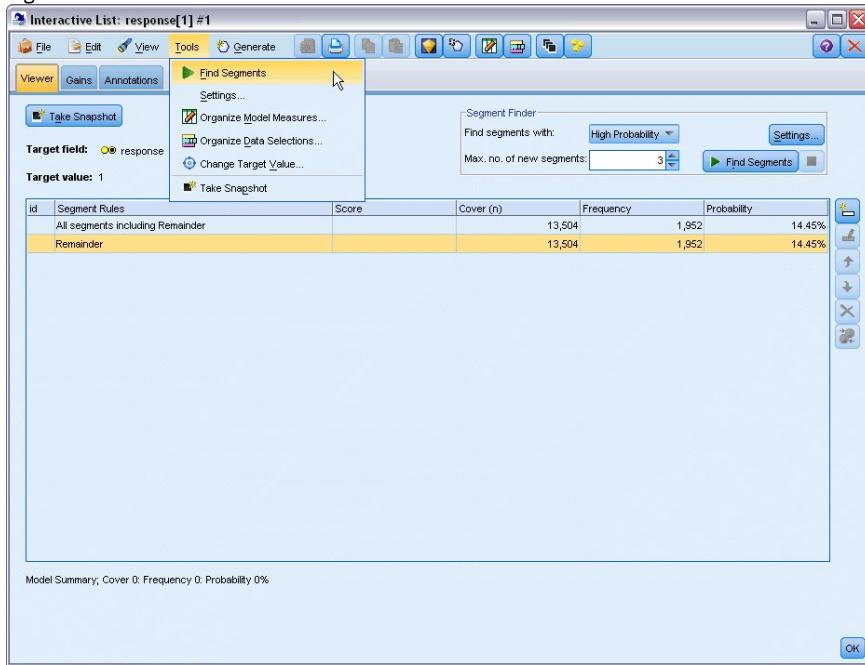
Figure 3. Interactive List viewer



Since no segments have yet been defined, all records fall under the remainder. Out of 13,504 records in the sample, 1,952 said Yes, for an overall hit rate of 14.45%. You want to improve on this rate by identifying segments of customers more (or less) likely to give a favorable response.

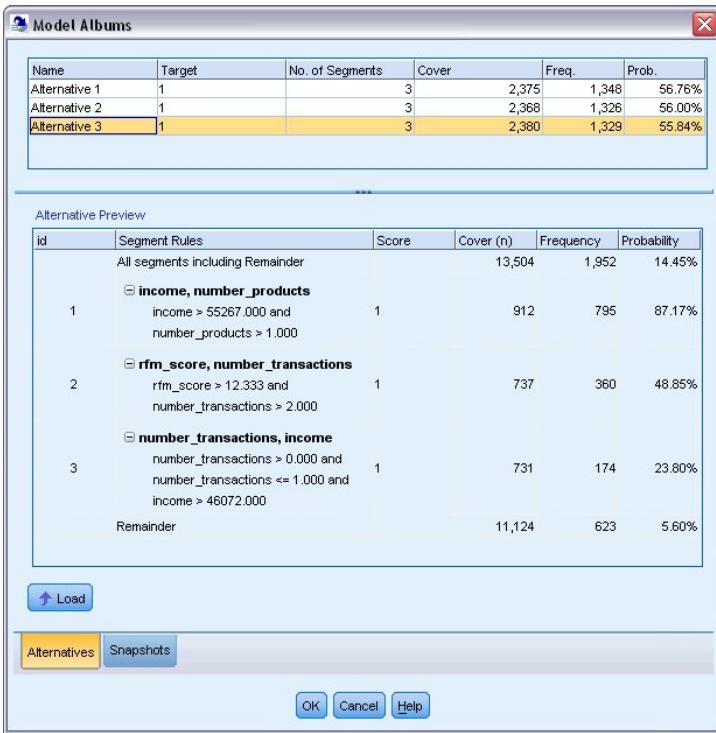
8. In the Interactive List viewer, from the menus choose:
Tools > Find Segments

Figure 4. Interactive List viewer



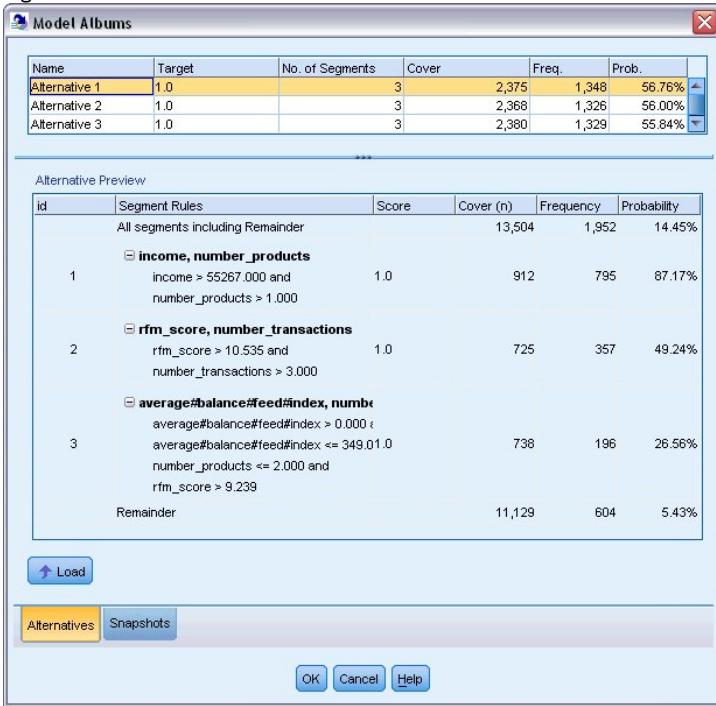
This runs the default mining task based on the settings you specified in the Decision List node. The completed task returns three alternative models, which are listed in the Alternatives tab of the Model Albums dialog box.

Figure 5. Available alternative models



9. Select the first alternative from the list; its details are shown in the Alternative Preview panel.

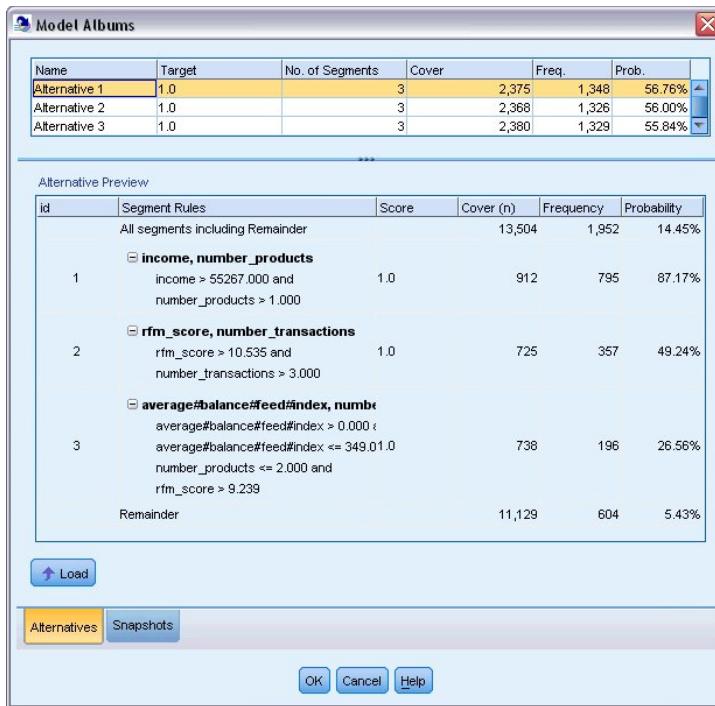
Figure 6. Alternative model selected



The Alternative Preview panel allows you to quickly browse any number of alternatives without changing the working model, making it easy to experiment with different approaches.

Note: To get a better look at the model, you may want to maximize the Alternative Preview panel within the dialog, as shown here. You can do this by dragging the panel border.

Figure 7. Alternative model selected



Using rules based on predictors, such as income, number of transactions per month, and RFM score, the model identifies segments with response rates that are higher than those for the sample overall. When the segments are combined, this model suggests that you could improve your hit rate to 56.76%. However, the model covers only a small portion of the overall sample, leaving over 11,000 records—with several hundred hits among them—to fall under the remainder. You want a model that will capture more of these hits while still excluding the low-performing segments.

10. To try a different modeling approach, from the menus choose:

Tools>Settings

Figure 8. Create/Edit Mining Task dialog box

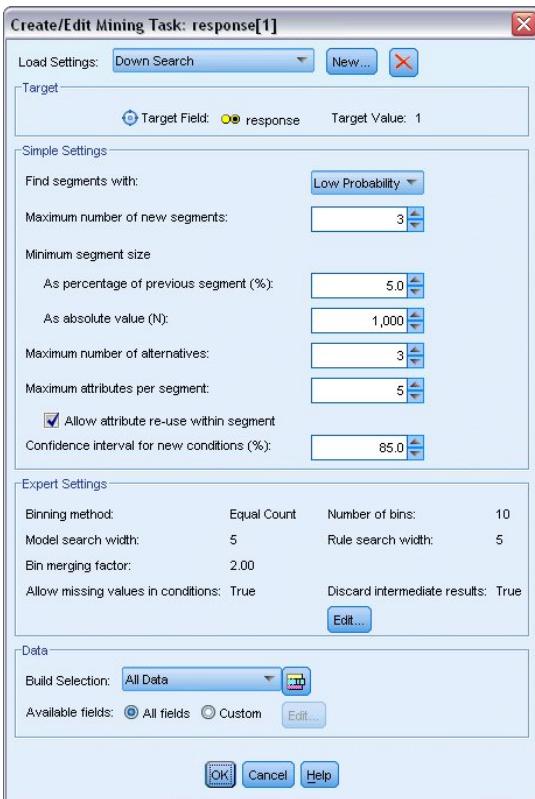
The screenshot shows the 'Create/Edit Mining Task: response[1]' dialog box. It has several sections:

- Load Settings:** dropdown set to 'response[1]', 'New...', and 'X' button.
- Target:** section with 'Target Field:' set to 'response' and 'Target Value:' set to '1'.
- Simple Settings:**
 - 'Find segments with:' dropdown set to 'High Probability'.
 - 'Maximum number of new segments:' dropdown set to '3'.
 - 'Minimum segment size' section with two options: 'As percentage of previous segment (%)' set to '5.0' and 'As absolute value (N)' set to '50'.
 - 'Maximum number of alternatives:' dropdown set to '3'.
 - 'Maximum attributes per segment:' dropdown set to '5'.
 - 'Allow attribute re-use within segment' checkbox.
 - 'Confidence interval for new conditions (%):' dropdown set to '85.0'.
- Expert Settings:**
 - 'Binning method:' dropdown set to 'Equal Count', 'Number of bins:' dropdown set to '10'.
 - 'Model search width:' dropdown set to '5', 'Rule search width:' dropdown set to '5'.
 - 'Bin merging factor:' dropdown set to '2.00'.
 - 'Allow missing values in conditions:' checkbox set to 'True'.
 - 'Discard intermediate results:' checkbox set to 'True'.
 - 'Edit...' button.
- Data:**
 - 'Build Selection:' dropdown set to 'All Data'.
 - 'Available fields:' section with 'All fields' radio button selected.
 - 'Edit...' button.

At the bottom are 'OK', 'Cancel', and 'Help' buttons.

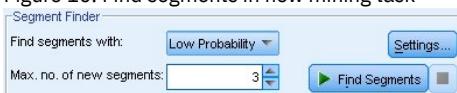
11. Click the New button (upper right corner) to create a second mining task, and specify Down Search as the task name in the New Settings dialog box.

Figure 9. Create/Edit Mining Task dialog box



12. Change the search direction to Low probability for the task. This will cause the algorithm to search for segments with the *lowest* response rates rather than the highest.
13. Increase the minimum segment size to 1,000. Click OK to return to the Interactive List viewer.
14. In Interactive List viewer, make sure that the *Segment Finder* panel is displaying the new task details and click Find Segments.

Figure 10. Find segments in new mining task



The task returns a new set of alternatives, which are displayed in the Alternatives tab of the Model Albums dialog box and can be previewed in the same manner as previous results.

Figure 11. Down Search model results

Model Albums

Name	Target	No. of Segments	Cover	Freq.	Prob.
Alternative 1	1	3	9,183	232	2.53%
Alternative 2	1	3	9,183	232	2.53%
Alternative 3	1	3	8,749	144	1.65%

Alternative Preview

id	Segment Rules	Score	Cover (n)	Frequency	Probability
1	All segments including Remainder		13,504	1,852	14.45%
1	months_customer months_customer = "0"	1	1,747	0	0.00%
2	rfm_score rfm_score <= 0.000	1	6,003	0	0.00%
3	income, rfm_score income > 40297.000 and income <= 55267.000 and rfm_score > 0.000 and rfm_score <= 10.535	1	1,433	232	16.19%
	Remainder		4,321	1,720	39.81%

Load

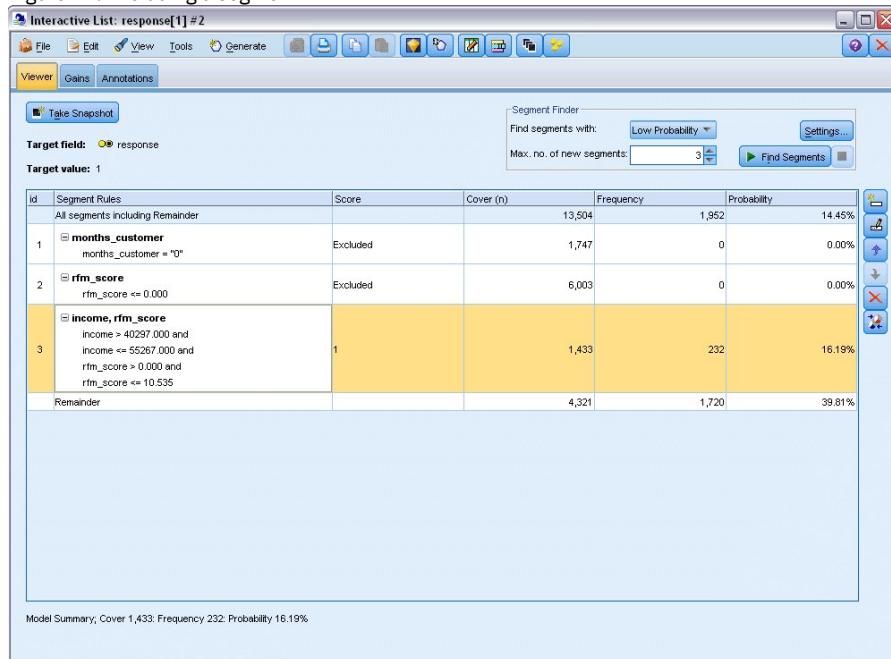
Alternatives Solutions

This time each model identifies segments with low response probabilities rather than high. Looking at the first alternative, simply excluding these segments will increase the hit rate for the remainder to 39.81%. This is lower than the model you looked at earlier but with higher coverage (meaning more total hits).

By combining the two approaches—using a Low Probability search to weed out uninteresting records, followed by a High Probability search—you may be able to improve this result.

- Click Load to make this (the first Down Search alternative) the working model and click OK to close the Model Albums dialog box.

Figure 12. Excluding a segment

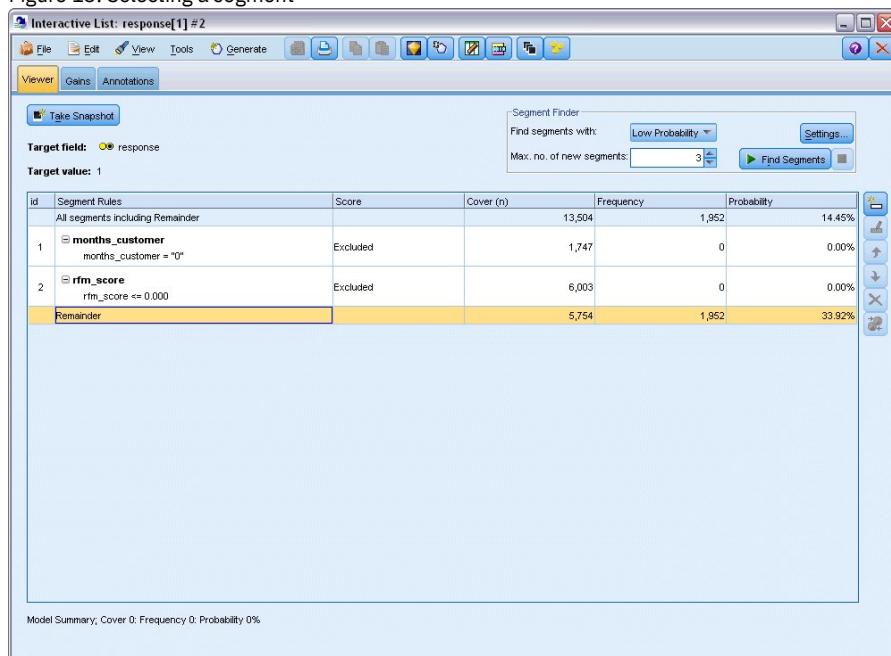


- Right-click on each of the first two segments and select Exclude Segment. Together, these segments capture almost 8,000 records with zero hits between them, so it makes sense to exclude them from future offers. (Excluded segments will be scored as null to indicate this.)
 - Right-click on the third segment and select Delete Segment. At 16.19%, the hit rate for this segment is not that different than the baseline rate of 14.45%, so it doesn't add enough information to justify keeping it in place.
- Note:* Deleting a segment is not the same as excluding it. Excluding a segment simply changes how it is scored, while deleting it removes it from the model entirely.

Having excluded the lowest-performing segments, you can now search for high-performing segments in the remainder.

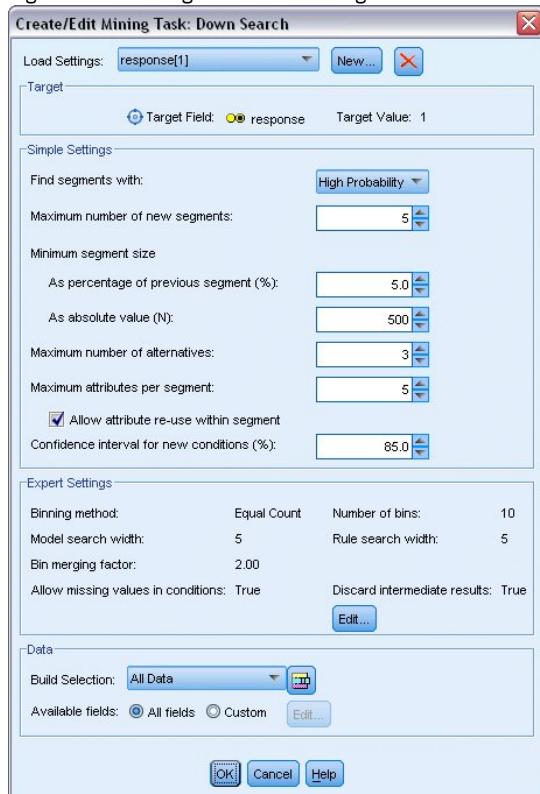
- Click on the remainder row in the table to select it, so that the next mining task will apply to the remainder only.

Figure 13. Selecting a segment



19. With the remainder selected, click Settings to reopen the Create/Edit Mining Task dialog box.
20. At the top, in Load Settings, select the default mining task: response[1].
21. Edit the Simple Settings to increase the number of new segments to 5 and the minimum segment size to 500.
22. Click OK to return to the Interactive List viewer.

Figure 14. Selecting the default mining task



23. Click Find Segments.

This displays yet another set of alternative models. By feeding the results of one mining task into another, these latest models contain a mix of high- and low-performing segments. Segments with low response rates are excluded, which means that they will be scored as null, while included segments will be scored as 1. The overall statistics reflect these exclusions, with the first alternative model showing a hit rate of 45.63%, with higher coverage (1,577 hits out of 3,456 records) than any of the previous models.

Figure 15. Alternatives for combined model

Name	Target	No. of Segments	Cover	Freq.	Prob.
Alternative 1	1	7	3,456	1,577	45.63%
Alternative 2	1	7	3,456	1,577	45.63%
Alternative 3	1	7	3,456	1,577	45.63%

ID	Segment Rules	Score	Cover (n)	Frequency	Probability
1	All segments including Remainder	13,504	1,952	14.45%	
2	months_customer = "0"	Excluded	1,747	0	0.00%
3	rfm_score <= 0.000	Excluded	6,003	0	0.00%
4	rfm_score > 12.333 and income > 52213.000	1	555	456	82.16%
5	income > 55267.000	1	643	551	85.69%
6	number_transactions, rfm_score > 2.000 and rfm_score > 12.333	1	533	206	38.65%

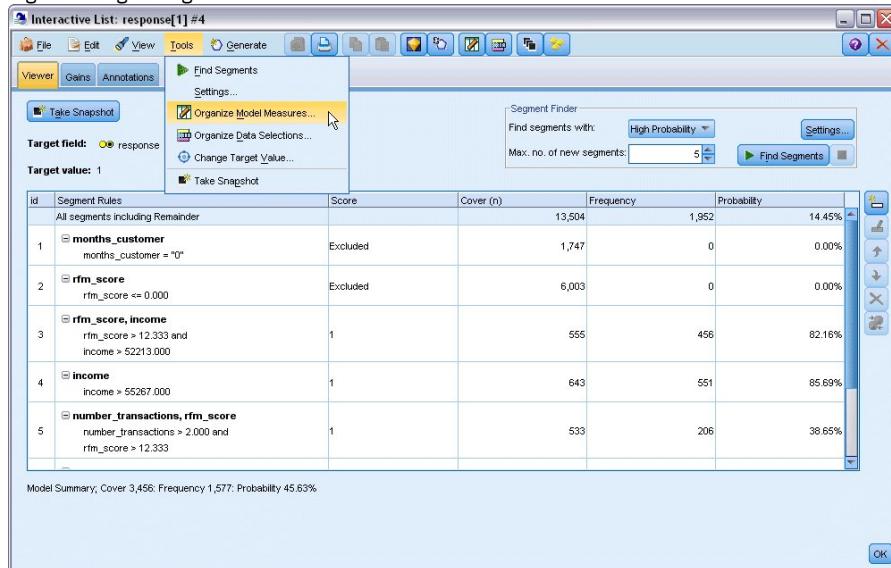
24. Preview the first alternative and then click Load to make it the working model.

[Next](#)

Calculating custom measures using Excel

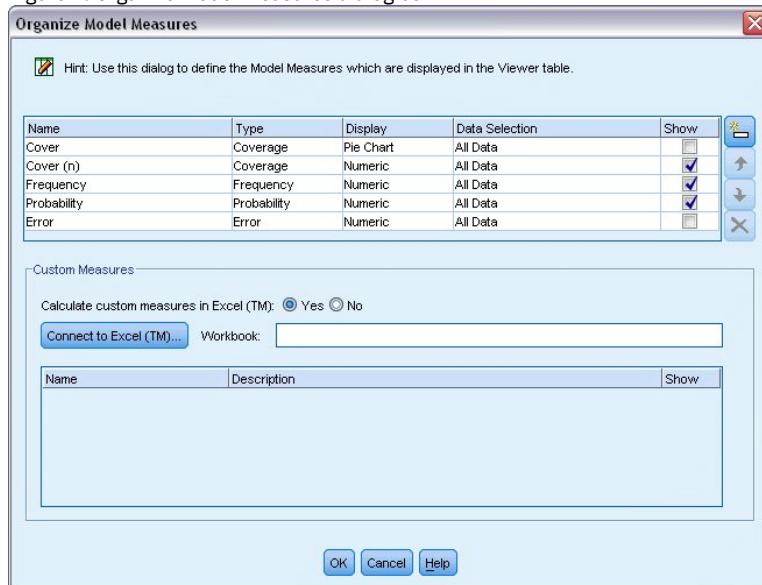
1. To gain a bit more insight as to how the model performs in practical terms, choose Organize Model Measures from the Tools menu.

Figure 1. Organizing model measures



The Organize Model Measures dialog box allows you to choose the measures (or columns) to show in the Interactive List viewer. You can also specify whether measures are computed against all records or a selected subset, and you can choose to display a pie chart rather than a number, where applicable.

Figure 2. Organize Model Measures dialog box



In addition, if you have Microsoft Excel installed, you can link to an Excel template that will calculate custom measures and add them to the interactive display.

2. In the Organize Model Measures dialog box, set Calculate custom measures in Excel (TM) to Yes.
3. Click Connect to Excel (TM)
4. Select the *template_profit.xlt* workbook, located under *streams* in the *Demos* folder of your IBM® SPSS® Modeler installation, and click Open to launch the spreadsheet.

Figure 3. Excel Model Measures worksheet

	#	Use	Metric: Frequency	Imported Metric: Cover	Calculated Metric: Profit Margin	Calculated Metric: Cumulative Profit	Target
	4	1				-2,500.00	
	5	2					

Model Measures / Settings / Configuration / Ready NUM

The Excel template contains three worksheets:

- Model Measures displays model measures imported from the model and calculates custom measures for export back to the model.
- Settings contains parameters to be used in calculating custom measures.
- Configuration defines the measures to be imported from and exported to the model.

The metrics exported back to the model are:

- **Profit Margin.** Net revenue from the segment
- **Cumulative Profit.** Total profit from campaign

As defined by the following formulas:

```
Profit Margin = Frequency * Revenue per respondent - Cover * Variable cost

Cumulative Profit = Total Profit Margin - Fixed cost
```

Note that Frequency and Cover are imported from the model.

The cost and revenue parameters are specified by the user on the Settings worksheet.

Figure 4. Excel Settings worksheet

12	Costs and revenue	
13	- Fixed costs	2,500.00
14	- Variable cost	0.50
15	- Revenue per respondent	100.00

Model Measures / Settings / Configuration / Ready NUM

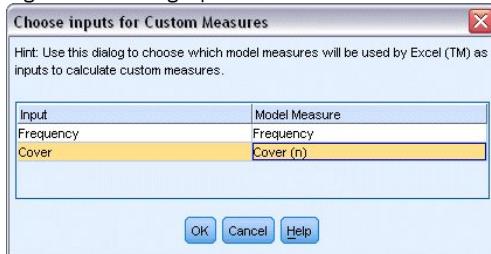
Fixed cost is the setup cost for the campaign, such as design and planning.

Variable cost is the cost of extending the offer to each customer, such as envelopes and stamps.

Revenue per respondent is the net revenue from a customer who responds to the offer.

5. To complete the link back to the model, use the Windows taskbar (or press Alt+Tab) to navigate back to the Interactive List viewer.

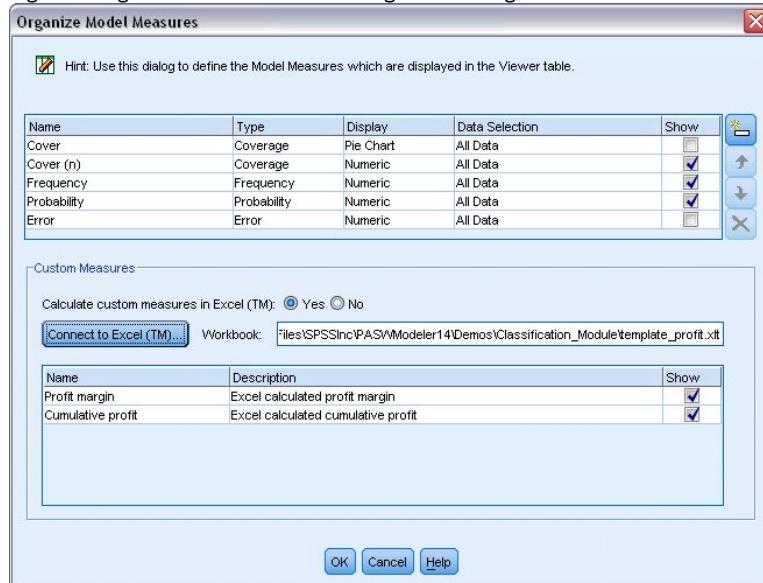
Figure 5. Choosing inputs for custom measures



The Choose Inputs for Custom Measures dialog box is displayed, allowing you to map inputs from the model to specific parameters defined in the template. The left column lists the available measures, and the right column maps these to spreadsheet parameters as defined in the Configuration worksheet.

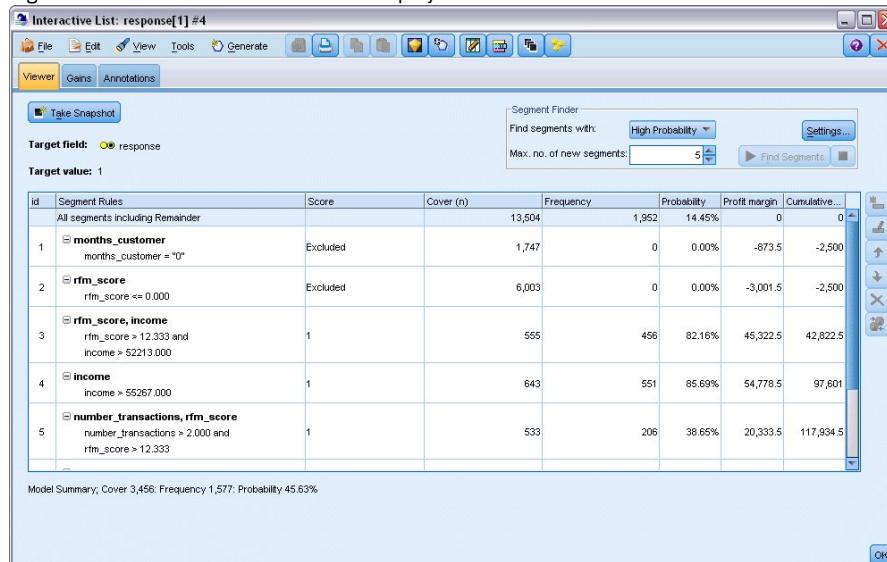
6. In the Model Measures column, select Frequency and Cover (n) against the respective inputs and click OK.
In this case, the parameter names in the template—Frequency and Cover (n)—happen to match the inputs, but different names could also be used.
7. Click OK in the Organize Model Measures dialog box to update the Interactive List viewer.

Figure 6. Organize Model Measures dialog box showing custom measures from Excel



The new measures are now added as new columns in the window and will be recalculated each time the model is updated.

Figure 7. Custom measures from Excel displayed in the Interactive List viewer



By editing the Excel template, any number of custom measures can be created.

[Next](#)

- [Modifying the Excel template](#)

Modifying the Excel template

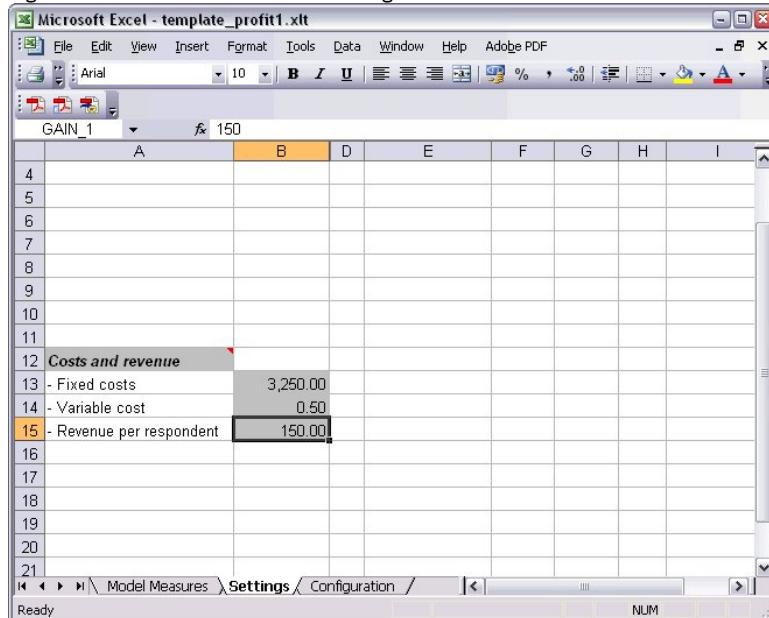
Although IBM® SPSS® Modeler is supplied with a default Excel template to use with the Interactive List viewer, you may want to change the settings or add your own. For example, the costs in the template may be incorrect for your organization and need amending.

Note: If you do modify an existing template, or create your own, remember to save the file with an Excel 2003 .xlt suffix.

To modify the default template with new cost and revenue details and update the Interactive List viewer with the new figures:

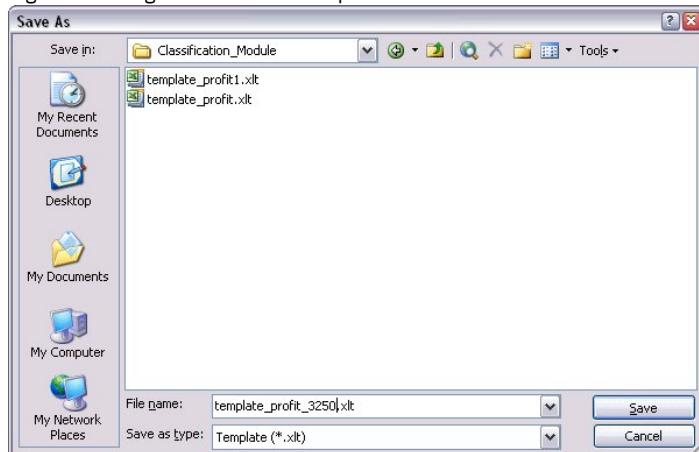
1. In the Interactive List viewer, choose Organize Model Measures from the Tools menu.
2. In the Organize Model Measures dialog box, click Connect to Excel™.
3. Select the *template_profit.xlt* workbook, and click Open to launch the spreadsheet.
4. Select the Settings worksheet.
5. Edit the Fixed costs to be 3,250.00, and the Revenue per respondent to be 150.00.

Figure 1. Modified values on Excel Settings worksheet



6. Save the modified template with a unique, relevant filename. Ensure it has an Excel 2003 .xlt extension.

Figure 2. Saving modified Excel template

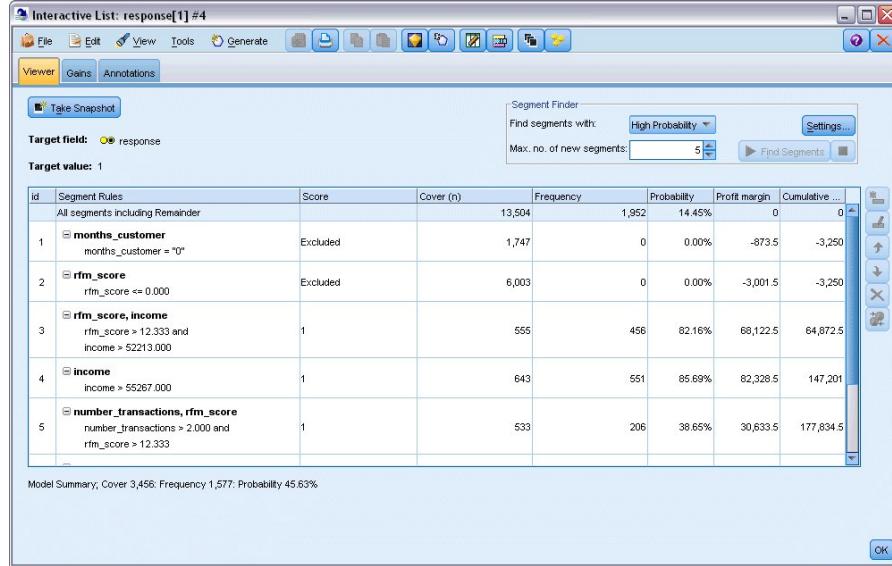


7. Use the Windows taskbar (or press Alt+Tab) to navigate back to the Interactive List viewer.
In the Choose Inputs for Custom Measures dialog box, select the measures you want to display and click OK.

8. In the Organize Model Measures dialog box, click OK to update the Interactive List viewer.

Obviously, this example has only shown one simple way of modifying the Excel template; you can make further changes that pull data from, and pass data to, the Interactive List viewer, or work within Excel to produce other output, such as graphs.

Figure 3. Modified custom measures from Excel displayed in the Interactive List viewer

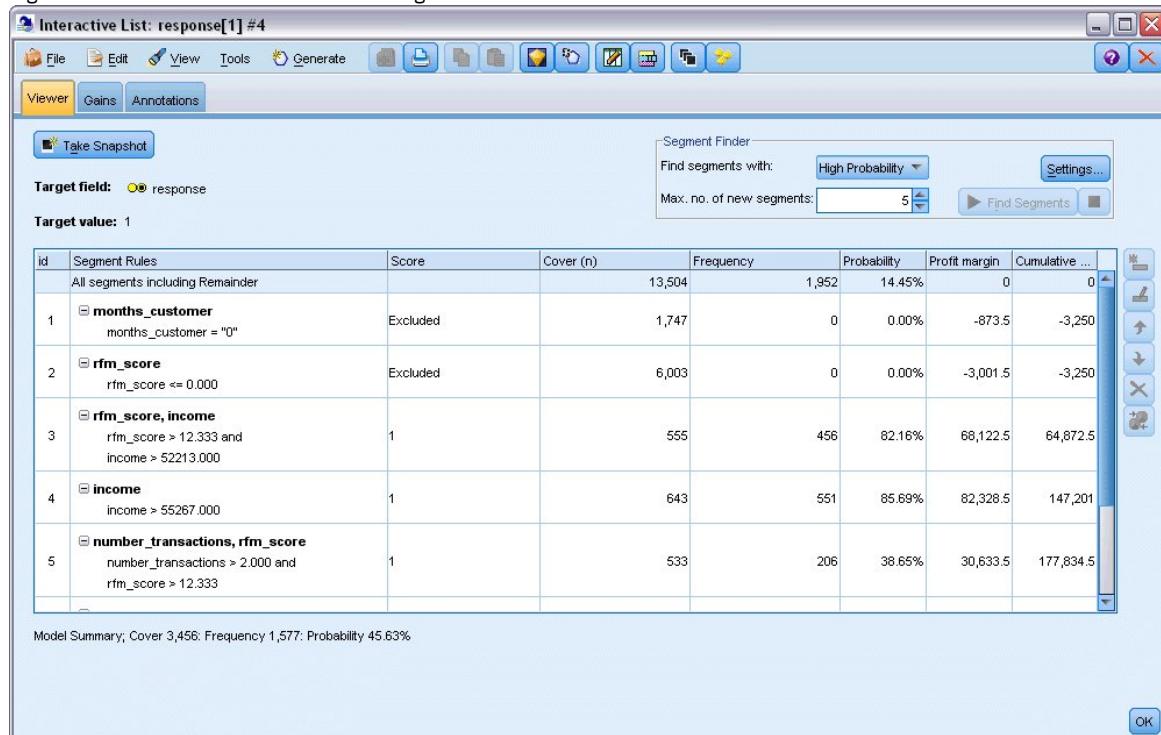


[Next](#)

Saving the Results

To save a model for later use during your interactive session, you can take a snapshot of the model, which will be listed on the Snapshots tab. You can return to any saved snapshot at any time during the interactive session.

Figure 1. Combined model with excluded segments



Continuing in this manner, you can experiment with additional mining tasks to search for additional segments. You can also edit existing segments, insert custom segments based on your own business rules, create data selections to optimize the model for specific groups, and customize the model in a number of other ways. Finally, you can explicitly include or exclude each segment as appropriate to specify how each will be scored.

When you are satisfied with your results, you can use the Generate menu to generate a model that can be added to streams or deployed for purposes of scoring.

Alternatively, to save the current state of your interactive session for another day, choose Update Modeling Node from the File menu. This will update the Decision List modeling node with the current settings, including mining tasks, model snapshots, data selections, and custom measures. The next time you run the stream, just make sure that Use saved session information is selected in the Decision List modeling node to restore the session to its current state.

Classifying Telecommunications Customers (Multinomial Logistic Regression)

Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric one.

For example, suppose a telecommunications provider has segmented its customer base by service usage patterns, categorizing the customers into four groups. If demographic data can be used to predict group membership, you can customize offers for individual prospective customers.

This example uses the stream named *telco_custcat.str*, which references the data file named *telco.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *telco_custcat.str* file is in the *streams* directory.

The example focuses on using demographic data to predict usage patterns. The target field *custcat* has four possible values that correspond to the four customer groups, as follows:

Value	Label
1	Basic Service
2	E-Service
3	Plus Service
4	Total Service

Because the target has multiple categories, a multinomial model is used. In the case of a target with two distinct categories, such as yes/no, true/false, or churn/don't churn, a binomial model could be created instead. See the topic [Telecommunications Churn \(Binomial Logistic Regression\)](#) for more information.

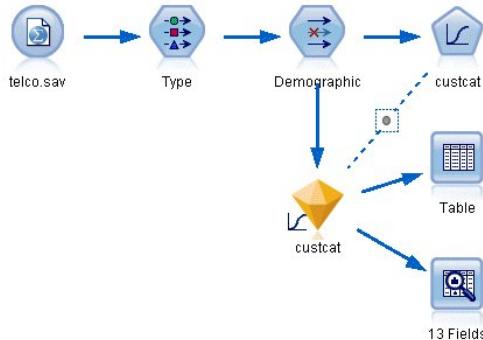
[Next](#)

- [Building the Stream](#)
- [Browsing the Model](#)

Building the Stream

1. Add a Statistics File source node pointing to *telco.sav* in the *Demos* folder.

Figure 1. Sample stream to classify customers using multinomial logistic regression



- a. Add a Type node and click Read Values, making sure that all measurement levels are set correctly. For example, most fields with values 0 and 1 can be regarded as flags.

Figure 2. Setting the measurement level for multiple fields

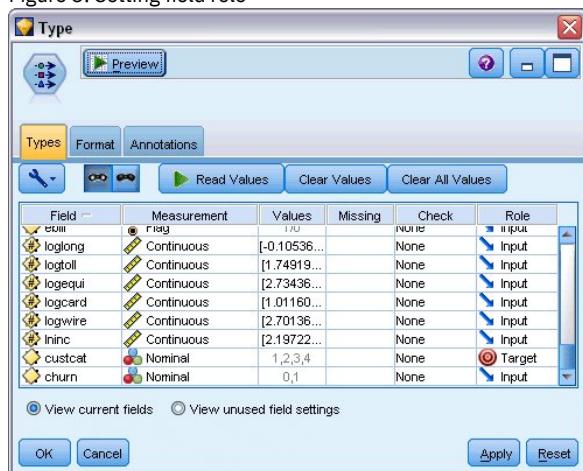


Tip: To change properties for multiple fields with similar values (such as 0/1), click the *Values* column header to sort fields by value, and then hold down the shift key while using the mouse or arrow keys to select all the fields you want to change. You can then right-click on the selection to change the measurement level or other attributes of the selected fields.

Notice that *gender* is more correctly considered as a field with a set of two values, instead of a flag, so leave its Measurement value as Nominal.

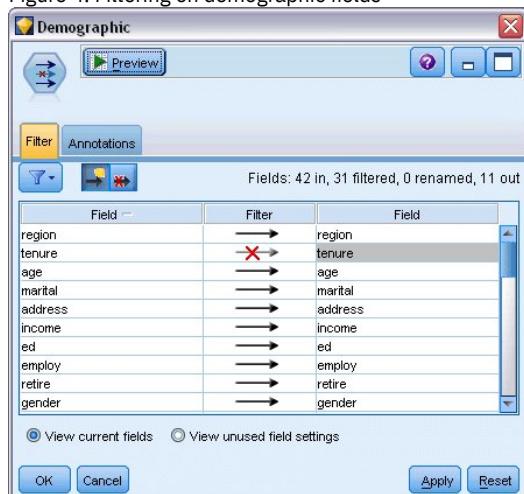
- Set the role for the *custcat* field to Target. All other fields should have their role set to Input.

Figure 3. Setting field role



Since this example focuses on demographics, use a Filter node to include only the relevant fields (*region*, *age*, *marital*, *address*, *income*, *ed*, *employ*, *retire*, *gender*, *reside*, and *custcat*). Other fields can be excluded for the purpose of this analysis.

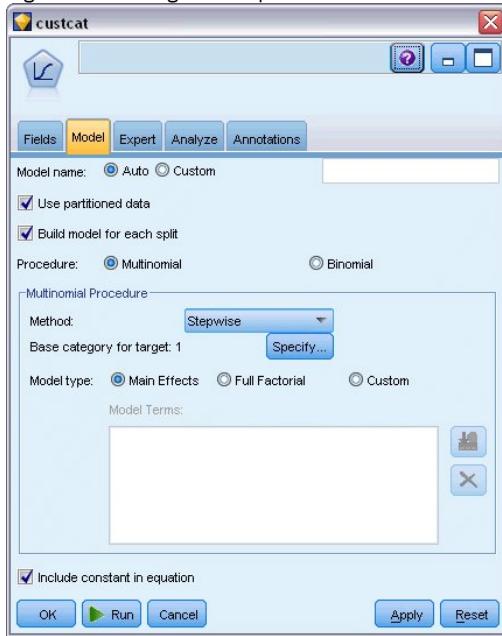
Figure 4. Filtering on demographic fields



(Alternatively, you could change the role to None for these fields rather than exclude them, or select the fields you want to use in the modeling node.)

2. In the Logistic node, click the Model tab and select the Stepwise method. Select Multinomial, Main Effects, and Include constant in equation as well.

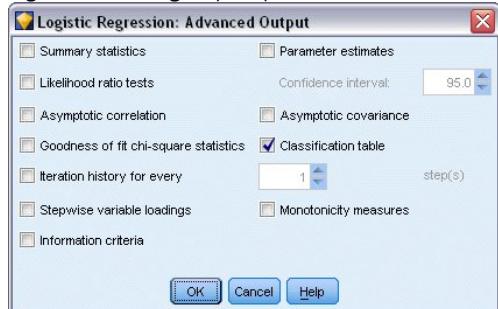
Figure 5. Choosing model options



Leave the Base category for target as 1. The model will compare other customers to those who subscribe to the Basic Service.

3. On the Expert tab, select the Expert mode, select Output, and, in the Advanced Output dialog box, select Classification table.

Figure 6. Choosing output options

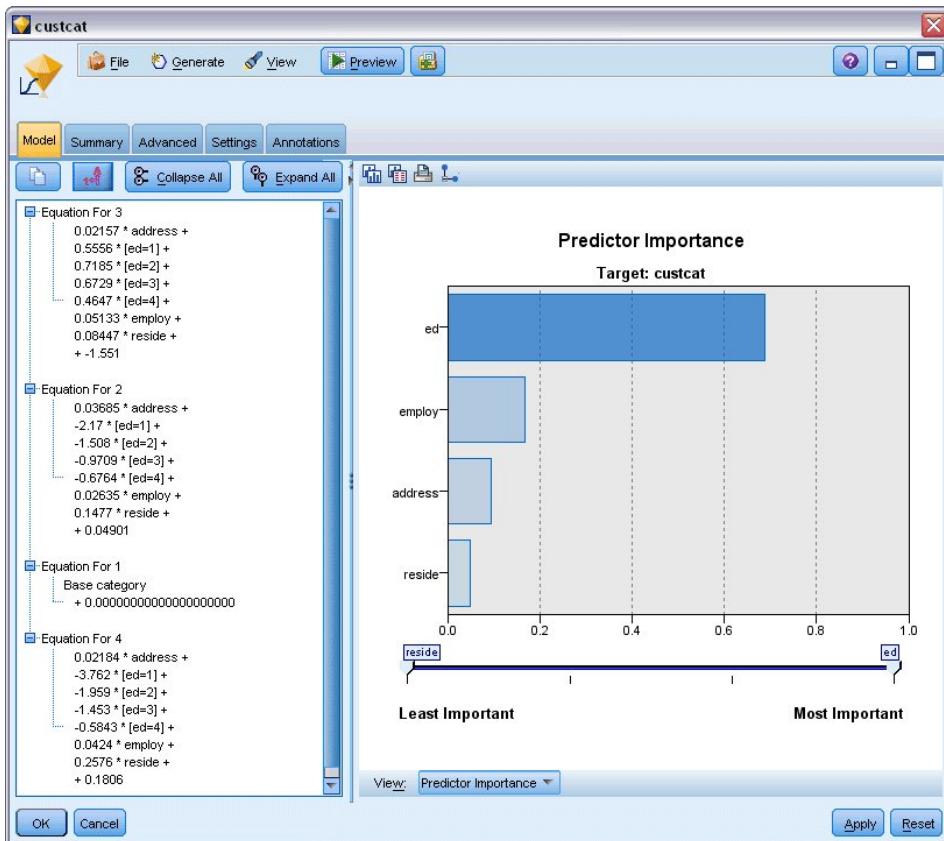


Browsing the Model

1. Execute the node to generate the model, which is added to the Models palette in the upper-right corner. To view its details, right-click on the generated model node and choose Browse.

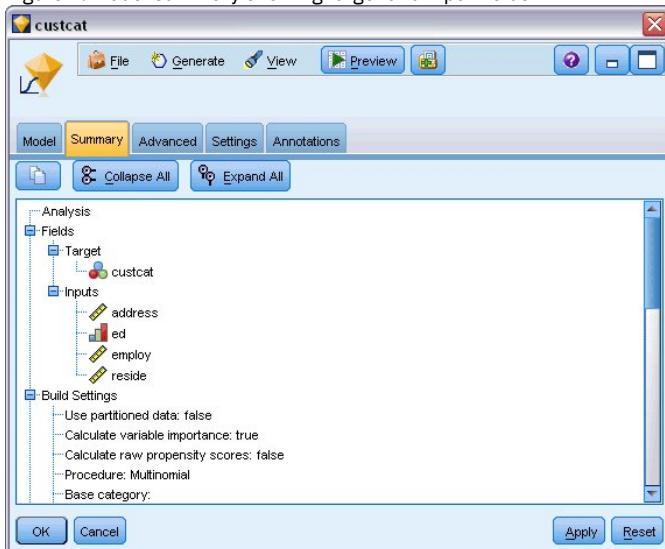
The model tab displays the equations used to assign records to each category of the target field. There are four possible categories, one of which is the base category for which no equation details are shown. Details are shown for the remaining three equations, where category 3 represents Plus Service, and so on.

Figure 1. Browsing the model results



The Summary tab shows (among other things) the target and inputs (predictor fields) used by the model. Note that these are the fields that were actually chosen based on the Stepwise method, not the complete list submitted for consideration.

Figure 2. Model summary showing target and input fields

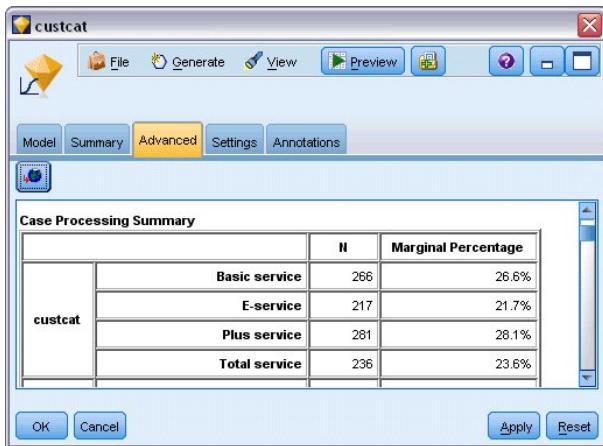


The items shown on the Advanced tab depend on the options selected on the Advanced Output dialog box in the modeling node.

One item that is always shown is the Case Processing Summary, which shows the percentage of records that falls into each category of the target field. This gives you a null model to use as a basis for comparison.

Without building a model that used predictors, your best guess would be to assign all customers to the most common group, which is the one for Plus service.

Figure 3. Case processing summary

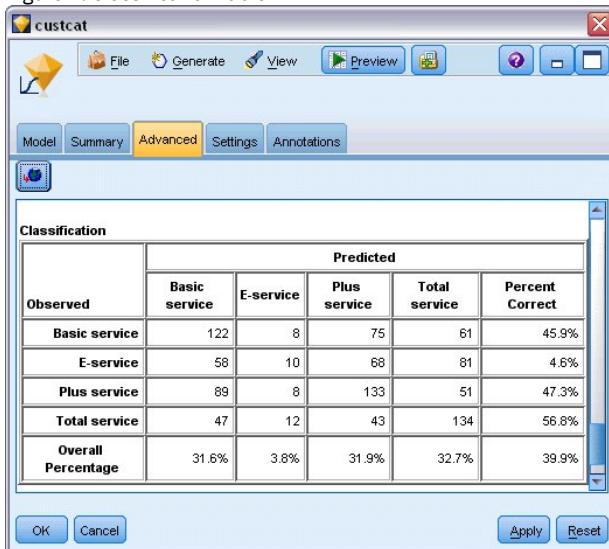


Based on the training data, if you assigned all customers to the null model, you would be correct $281/1000 = 28.1\%$ of the time. The Advanced tab contains further information that enables you to examine the model's predictions. You can then compare the predictions with the null model's results to see how well the model works with your data.

At the bottom of the Advanced tab, the Classification table shows the results for your model, which is correct 39.9% of the time.

In particular, your model excels at identifying Total Service customers (category 4) but does a very poor job of identifying E-service customers (category 2). If you want better accuracy for customers in category 2, you may need to find another predictor to identify them.

Figure 4. Classification table



Depending on what you want to predict, the model may be perfectly adequate for your needs. For example, if you are not concerned with identifying customers in category 2, the model may be accurate enough for you. This may be the case where the E-service is a loss-leader that brings in little profit.

If, for example, your highest return on investment comes from customers who fall into category 3 or 4, the model may give you the information you need.

To assess how well the model actually fits the data, a number of diagnostics are available in the Advanced Output dialog box when you are building the model. Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the *IBM SPSS Modeler Algorithms Guide*, available from the |Documentation directory of the installation disk.

Note also that these results are based on the training data only. To assess how well the model generalizes to other data in the real world, you can use a Partition node to hold out a subset of records for purposes of testing and validation.

Telecommunications Churn (Binomial Logistic Regression)

Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric one.

This example uses the stream named `telco_churn.str`, which references the data file named `telco.sav`. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The `telco_churn.str` file is in the *streams* directory.

For example, suppose a telecommunications provider is concerned about the number of customers it is losing to competitors. If service usage data can be used to predict which customers are liable to transfer to another provider, offers can be customized to retain as many customers as possible.

This example focuses on using usage data to predict customer loss (churn). Because the target has two distinct categories, a binomial model is used. In the case of a target with multiple categories, a multinomial model could be created instead. See the topic [Classifying Telecommunications Customers \(Multinomial Logistic Regression\)](#) for more information.

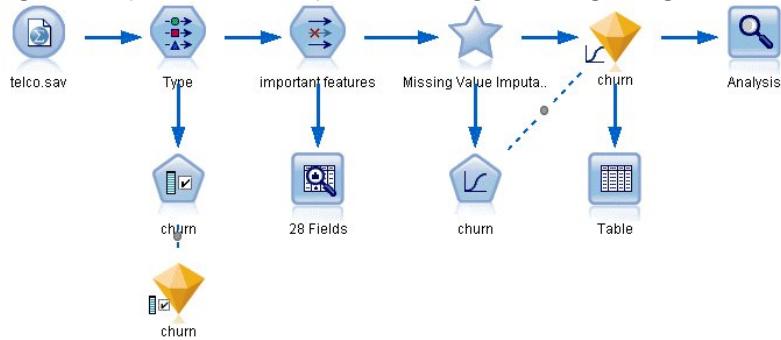
[Next](#)

- [Building the Stream](#)
- [Browsing the Model](#)

Building the Stream

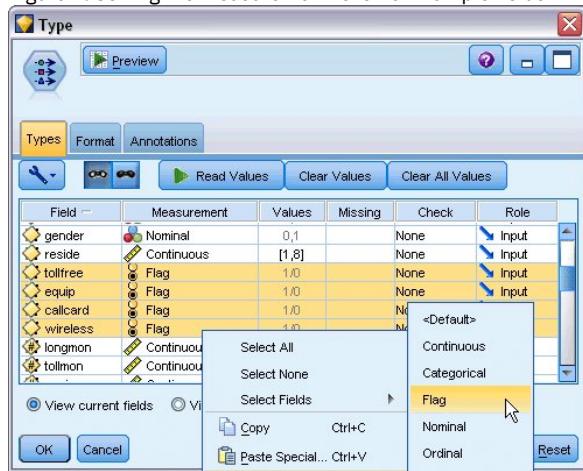
1. Add a Statistics File source node pointing to `telco.sav` in the *Demos* folder.

Figure 1. Sample stream to classify customers using binomial logistic regression



2. Add a Type node to define fields, making sure that all measurement levels are set correctly. For example, most fields with values 0 and 1 can be regarded as flags, but certain fields, such as gender, are more accurately viewed as a nominal field with two values.

Figure 2. Setting the measurement level for multiple fields



Tip: To change properties for multiple fields with similar values (such as 0/1), click the *Values* column header to sort fields by value, and then hold down the Shift key while using the mouse or arrow keys to select all of the fields that you want to change. You can then right-click on the selection to change the measurement level or other attributes of the selected fields.

3. Set the measurement level for the `churn` field to Flag, and set the role to Target. All other fields should have their role set to Input.

Figure 3. Setting the measurement level and role for the `churn` field



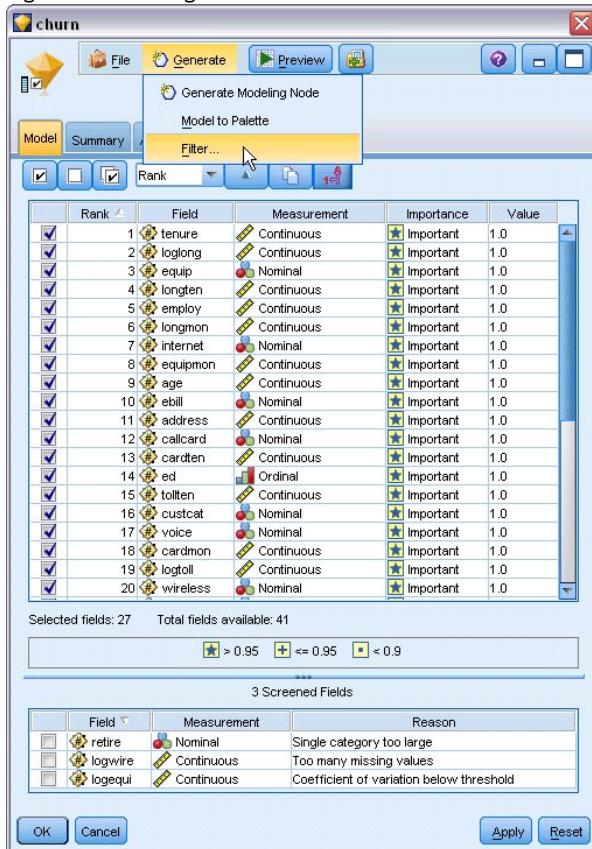
4. Add a Feature Selection modeling node to the Type node.

Using a Feature Selection node enables you to remove predictors or data that do not add any useful information with respect to the predictor/target relationship.

5. Run the stream.

6. Open the resulting model nugget, and from the Generate menu, choose Filter to create a Filter node.

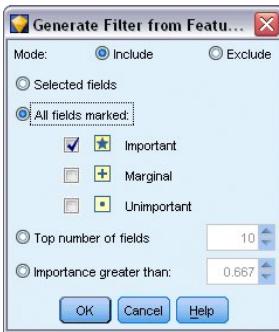
Figure 4. Generating a Filter node from a Feature Selection node



Not all of the data in the *telco.sav* file will be useful in predicting churn. You can use the filter to only select data considered to be important for use as a predictor.

7. In the Generate Filter dialog box, select All fields marked: Important and click OK.
8. Attach the generated Filter node to the Type node.

Figure 5. Selecting important fields



9. Attach a Data Audit node to the generated Filter node.

Open the Data Audit node and click Run.

10. On the Quality tab of the Data Audit browser, click the % Complete column to sort the column by ascending numerical order. This lets you identify any fields with large amounts of missing data; in this case the only field you need to amend is *logtoll*, which is less than 50% complete.

11. In the *Impute Missing* column for *logtoll*, click Specify.

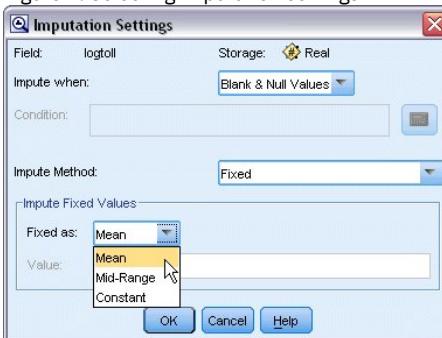
Figure 6. Imputing missing values for logtoll

Field	Measurement	Outliers	Extremes	Action	Impute Missing	Method	% Complete	Valid
logtoll	Continuous	2	0 None	Never	Fixed	47.5	96.43%	100
tenure	Continuous	0	0 None	Never	Fixed	100		
age	Continuous	0	0 None	Blank Values	Fixed	100		
address	Continuous	12	0 None	Null Values	Fixed	100		
income	Continuous	9	6 None	Blank & Null Value	Fixed	100		
ed	Ordinal	--	--	Condition...	Fixed	100		
employ	Continuous	8	0 None	Condition...	Fixed	100		
equip	Flag	--	--	Specify...	Never	100		
callcard	Flag	--	--	Never	Fixed	100		
wireless	Flag	--	--	Never	Fixed	100		
longmon	Continuous	18	4 None	Never	Fixed	100		
tollmon	Continuous	9	1 None	Never	Fixed	100		
equipment	Continuous	2	0 None	Never	Fixed	100		
cardmon	Continuous	11	3 None	Never	Fixed	100		
wiremon	Continuous	8	1 None	Never	Fixed	100		
longten	Continuous	20	4 None	Never	Fixed	100		
tolten	Continuous	18	2 None	Never	Fixed	100		
cardten	Continuous	11	6 None	Never	Fixed	100		
voice	Flag	--	--	Never	Fixed	100		

12. For Impute when, select Blank and Null values. For Fixed As, select Mean and click OK.

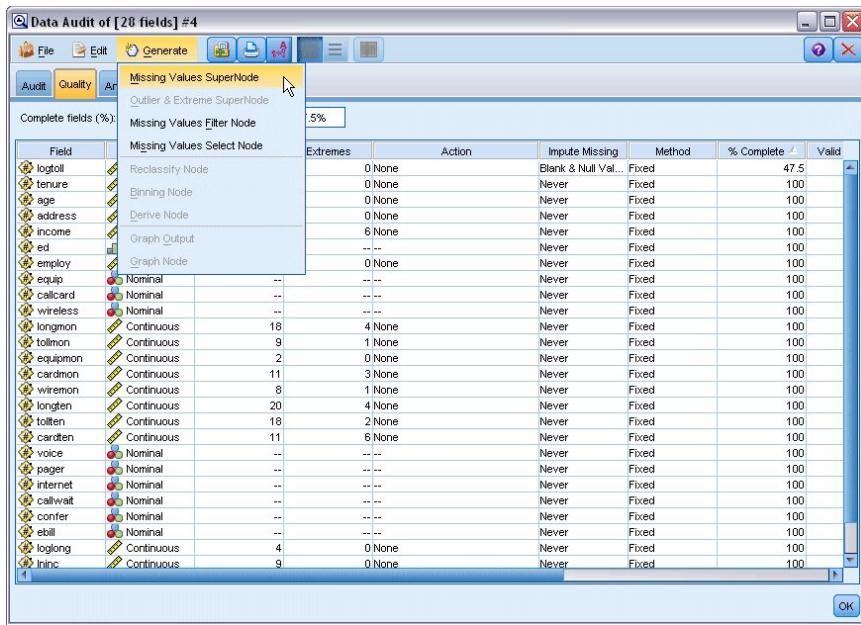
Selecting Mean ensures that the imputed values do not adversely affect the mean of all values in the overall data.

Figure 7. Selecting imputation settings



13. On the Data Audit browser Quality tab, generate the Missing Values SuperNode. To do this, from the menus choose: Generate > Missing Values SuperNode

Figure 8. Generating a missing values SuperNode

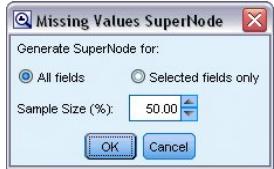


In the Missing Values SuperNode dialog box, increase the Sample Size to 50% and click OK.

The SuperNode is displayed on the stream canvas, with the title: *Missing Value Imputation*.

14. Attach the SuperNode to the Filter node.

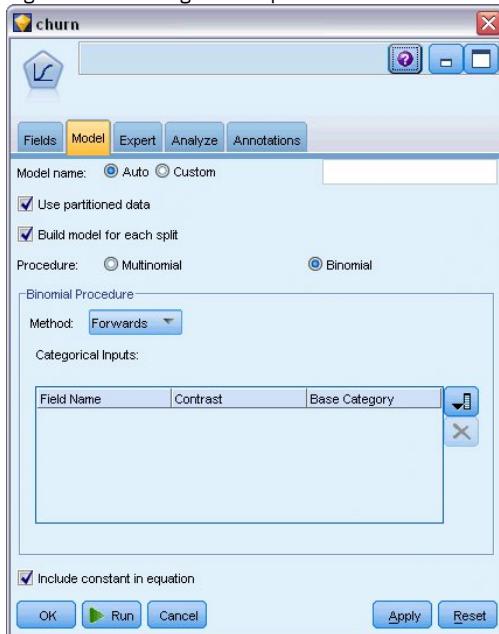
Figure 9. Specifying sample size



15. Add a Logistic node to the SuperNode.

16. In the Logistic node, click the Model tab and select the Binomial procedure. In the *Binomial Procedure* area, select the Forwards method.

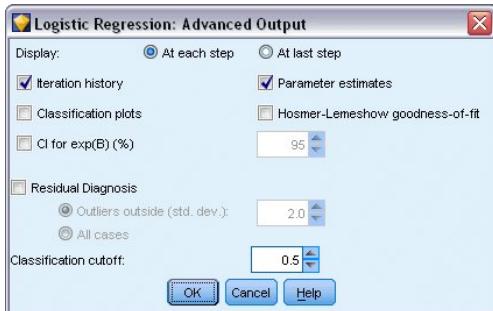
Figure 10. Choosing model options



17. On the Expert tab, select the Expert mode and then click Output. The Advanced Output dialog box is displayed.

18. In the Advanced Output dialog, select At each step as the *Display* type. Select Iteration history and Parameter estimates and click OK.

Figure 11. Choosing output options



[Next](#)

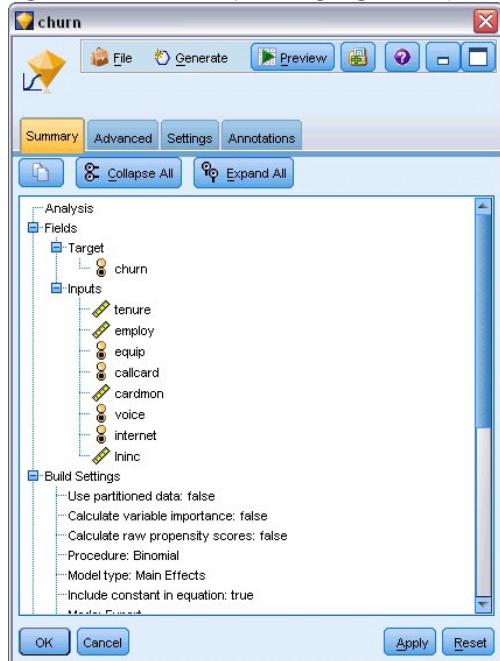
Browsing the Model

1. On the Logistic node, click Run to create the model.

The model nugget is added to the stream canvas, and also to the Models palette in the upper-right corner. To view its details, right-click on the model nugget and select Edit or Browse.

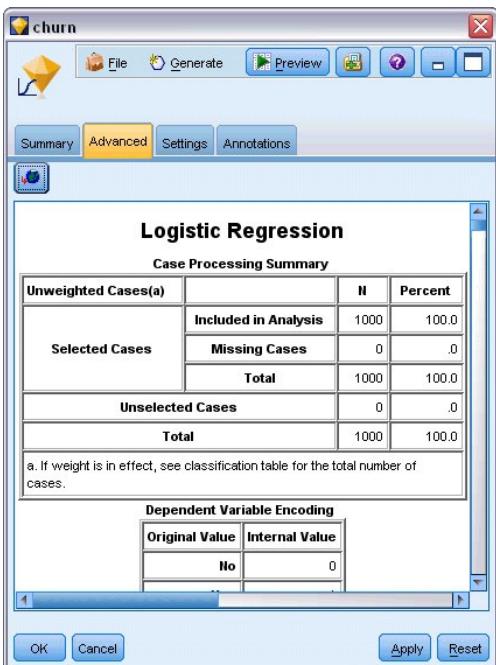
The Summary tab shows (among other things) the target and inputs (predictor fields) used by the model. Note that these are the fields that were actually chosen based on the Forwards method, not the complete list submitted for consideration.

Figure 1. Model summary showing target and input fields



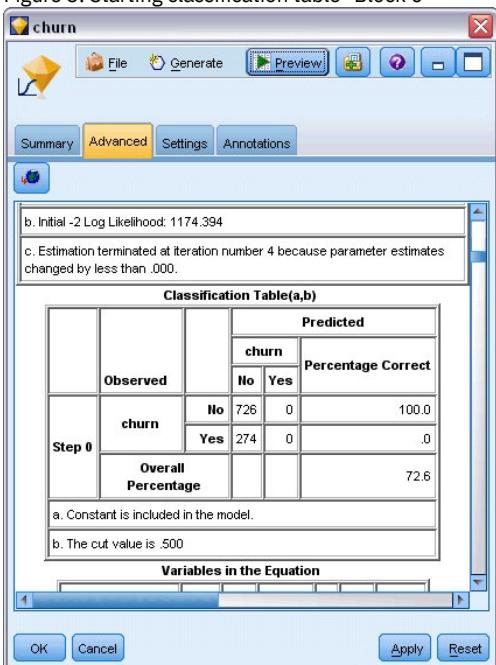
The items shown on the Advanced tab depend on the options selected on the Advanced Output dialog box in the Logistic node. One item that is always shown is the Case Processing Summary, which shows the number and percentage of records included in the analysis. In addition, it lists the number of missing cases (if any) where one or more of the input fields are unavailable and any cases that were not selected.

Figure 2. Case processing summary



2. Scroll down from the Case Processing Summary to display the Classification Table under Block 0: Beginning Block. The Forward Stepwise method starts with a null model - that is, a model with no predictors - that can be used as a basis for comparison with the final built model. The null model, by convention, predicts everything as a 0, so the null model is 72.6% accurate simply because the 726 customers who didn't churn are predicted correctly. However, the customers who did churn aren't predicted correctly at all.

Figure 3. Starting classification table- Block 0



3. Now scroll down to display the Classification Table under Block 1: Method = Forward Stepwise. This Classification Table shows the results for your model as a predictor is added in at each of the steps. Already, in the first step - after just one predictor has been used - the model has increased the accuracy of the churn prediction from 0.0% to 29.9%

Figure 4. Classification table - Block 1

Classification Table(a)

Observed		Predicted		Percentage Correct	
		churn			
		No	Yes		
Step 1	churn	668	58	92.0	
	Yes	192	82	29.9	
	Overall Percentage			75.0	
Step 2	churn	657	69	90.5	
	Yes	160	114	41.6	
	Overall Percentage			77.1	
Step 3	churn	661	65	91.0	
	Yes	153	121	44.2	
	Overall Percentage			79.1	

OK Cancel Apply Reset

4. Scroll down to the bottom of this Classification Table.

The Classification Table shows that the last step is step 8. At this stage the algorithm has decided that it no longer needs to add any further predictors into the model. Although the accuracy of the non-churning customers has decreased a little to 91.2%, the accuracy of the prediction for those who did churn has risen from the original 0% to 47.1%. This is a significant improvement over the original null model that used no predictors.

Figure 5. Classification table - Block 1

Step 7

Overall Percentage				
churn	No	657	69	90.5
Yes	144	130		47.4
Overall Percentage				78.7

Step 8

Overall Percentage				
churn	No	662	64	91.2
Yes	145	129		47.1
Overall Percentage				79.1

a. The cut value is .500

Variables in the Equation

	B	S.E.	Wald	df	Sig.	Exp(B)	
Step 1(a)	tenure	-.046	.004	123.346	1	.000	.955
Step 1(a)	Constant	462	136	11.574	1	.001	1.587

OK Cancel Apply Reset

For a customer who wants to reduce churn, being able to reduce it by nearly half would be a major step in protecting their income streams.

Note: This example also shows how taking the Overall Percentage as a guide to a model's accuracy may, in some cases, be misleading. The original null model was 72.6% accurate overall, whereas the final predicted model has an overall accuracy of 79.1%; however, as we have seen, the accuracy of the actual individual category predictions were vastly different.

To assess how well the model actually fits the data, a number of diagnostics are available in the Advanced Output dialog box when you are building the model. Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the *IBM SPSS Modeler Algorithms Guide*, available from the |Documentation directory of the installation disk.

Note also that these results are based on the training data only. To assess how well the model generalizes to other data in the real world, you would use a Partition node to hold out a subset of records for purposes of testing and validation.

Forecasting Bandwidth Utilization (Time Series)

- [Forecasting with the Time Series Node](#)
- [Reapplying a Time Series Model](#)

Forecasting with the Time Series Node

An analyst for a national broadband provider is required to produce forecasts of user subscriptions in order to predict utilization of bandwidth. Forecasts are needed for each of the local markets that make up the national subscriber base. You will use time series modeling to produce forecasts for the next three months for a number of local markets. A second example shows how you can convert source data if it is not in the correct format for input to the Time Series node.

These examples use the stream named *broadband_create_models.str*, which references the data file named *broadband_1.sav*. These files are available from the *Demos* folder of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *broadband_create_models.str* file is in the *streams* folder.

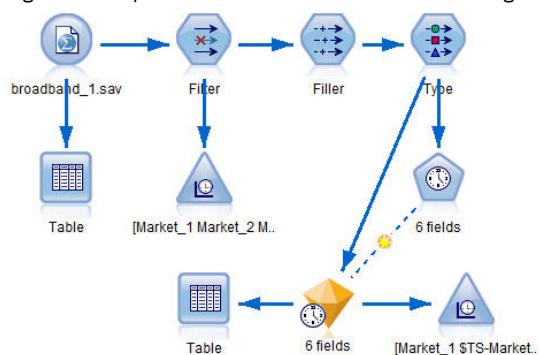
The last example demonstrates how to apply the saved models to an updated dataset in order to extend the forecasts by another three months.

In IBM SPSS Modeler, you can produce multiple time series models in a single operation. The source file you'll be using has time series data for 85 different markets, although for the sake of simplicity you will only model five of these markets, plus the total for all markets.

The *broadband_1.sav* data file has monthly usage data for each of 85 local markets. For the purposes of this example, only the first five series will be used; a separate model will be created for each of these five series, plus a total.

The file also includes a date field that indicates the month and year for each record. This field will be used to label records. The date field reads into IBM SPSS Modeler as a string, but in order to use the field in IBM SPSS Modeler you will convert the storage type to numeric Date format using a Filler node.

Figure 1. Sample stream to show Time Series modeling



The Time Series node requires that each series be in a separate column, with a row for each interval. IBM SPSS Modeler provides methods for transforming data to match this format if necessary.

Figure 2. Monthly subscription data for broadband local markets

A screenshot of the SPSS Modeler interface showing a table window titled "Table (89 fields, 60 records)". The table contains 60 rows of data with 89 fields each. The first row is highlighted in yellow. The columns are labeled Market_1, Market_2, Market_3, Market_4, Market_5, Market_6, Market_7, Market_8, and Mar.

[Next](#)

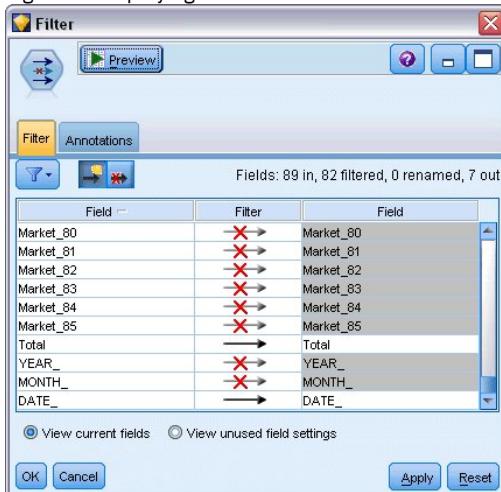
- [Creating the Stream](#)
- [Examining the Data](#)
- [Defining the Dates](#)
- [Defining the Targets](#)
- [Setting the Time Intervals](#)
- [Creating the Model](#)
- [Examining the model](#)
- [Summary](#)

Creating the Stream

1. Create a new stream and add a Statistics File source node pointing to *broadband_1.sav*.
2. Use a Filter node to filter out the *Market_6* to *Market_85* fields and the *MONTH_* and *YEAR_* fields to simplify the model.

Tip: To select multiple adjacent fields in a single operation, click the *Market_6* field, hold down the left mouse button and drag the mouse down to the *Market_85* field. Selected fields are highlighted in blue. To add the other fields, hold down the Ctrl key and click the *MONTH_* and *YEAR_* fields.

Figure 1. Simplifying the model

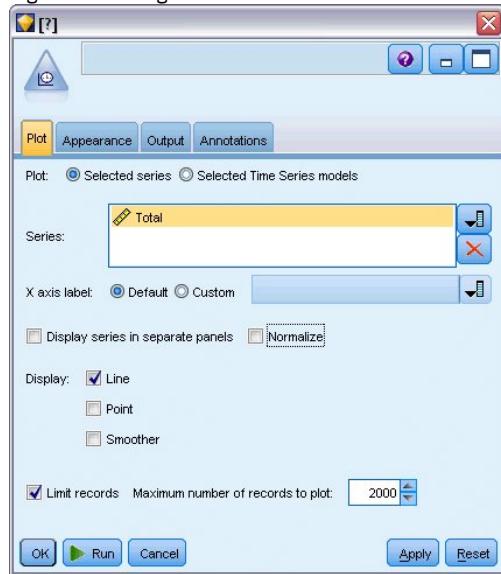


[Next](#)

Examining the Data

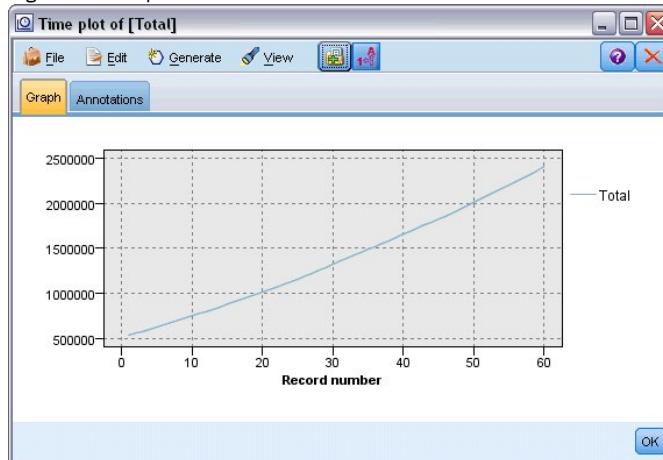
It is always a good idea to have a feel for the nature of your data before building a model. Do the data exhibit seasonal variations? Although the Expert Modeler can automatically find the best seasonal or nonseasonal model for each series, you can often obtain faster results by limiting the search to nonseasonal models when seasonality is not present in your data. Without examining the data for each of the local markets, we can get a rough picture of the presence or absence of seasonality by plotting the total number of subscribers over all five markets.

Figure 1. Plotting the total number of subscribers



1. From the Graphs palette, attach a Time Plot node to the Filter node.
2. Add the *Total* field to the Series list.
3. Deselect the Display series in separate panels and Normalize check boxes.
4. Click Run.

Figure 2. Time plot of Total field

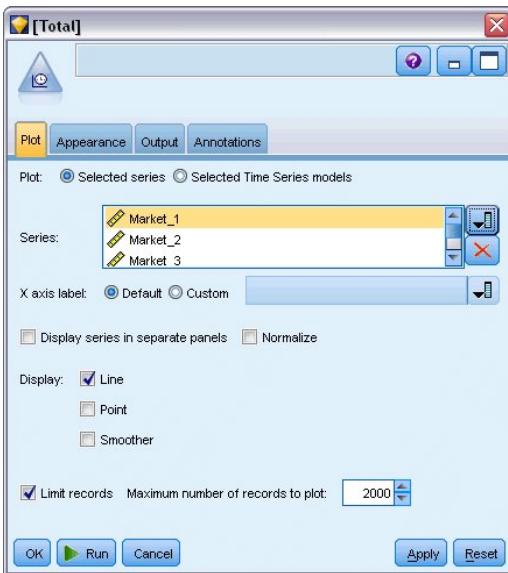


The series exhibits a very smooth upward trend with no hint of seasonal variations. There might be individual series with seasonality, but it appears that seasonality is not a prominent feature of the data in general.

Of course you should inspect each of the series before ruling out seasonal models. You can then separate out series exhibiting seasonality and model them separately.

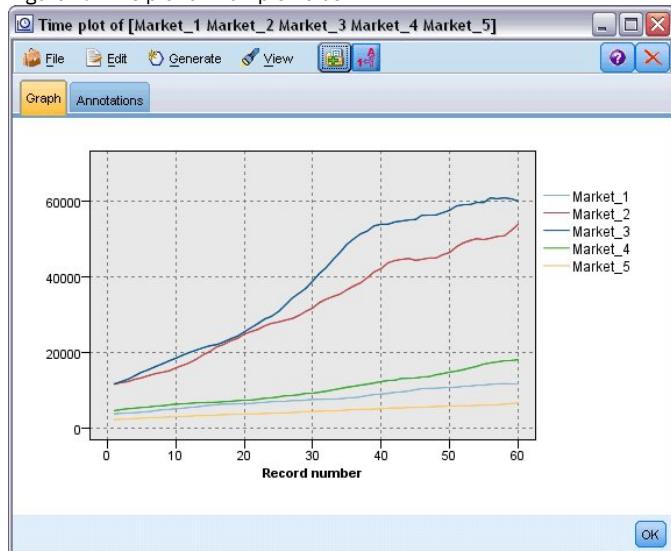
IBM® SPSS® Modeler makes it easy to plot multiple time series together.

Figure 3. Plotting multiple time series



5. Reopen the Time Plot node.
6. Remove the *Total* field from the Series list (select it and click the red X button).
7. Add the *Market_1* through *Market_5* fields to the list.
8. Click Run.

Figure 4. Time plot of multiple fields



Inspection of each of the markets reveals a steady upward trend in each case. Although some markets are a little more erratic than others, there is no evidence of seasonality to be seen.

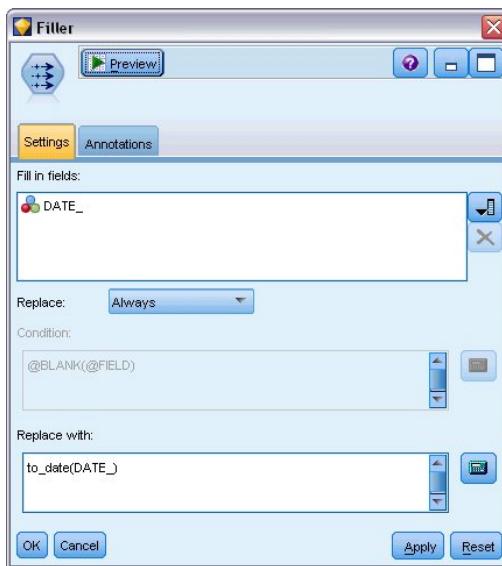
[Next](#)

Defining the Dates

Now you need to change the storage type of the *DATE_* field to Date format.

1. Attach a Filler node to the Filter node.
2. Open the Filler node and click the field selector button.
3. Select *DATE_* to add it to Fill in fields.
4. Set the Replace condition to Always.
5. Set the value of Replace with to `to_date(DATE_)`.

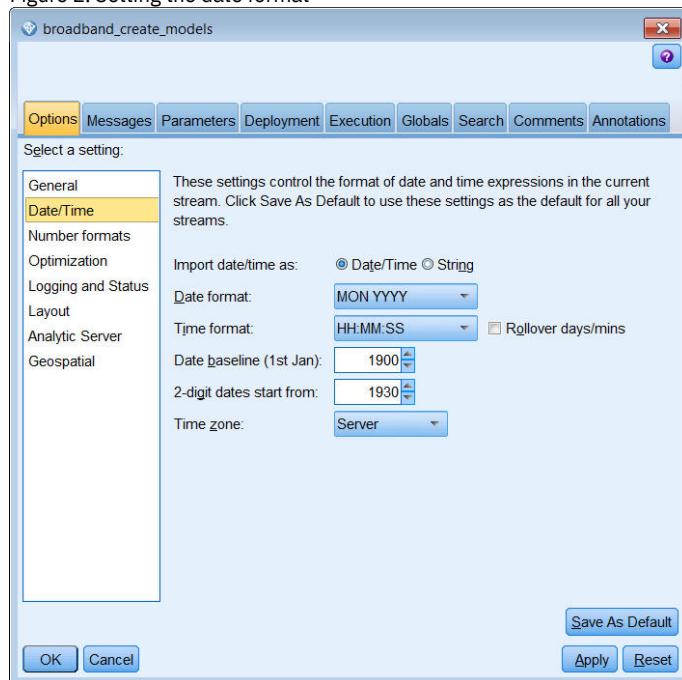
Figure 1. Setting the date storage type



Change the default date format to match the format of the Date field. This is necessary for the conversion of the Date field to work as expected.

6. On the menu, choose Tools > Stream Properties > Options to display the Stream Options dialog box.
7. Select the Date/Time pane and set the default Date format to MON YYYY .

Figure 2. Setting the date format

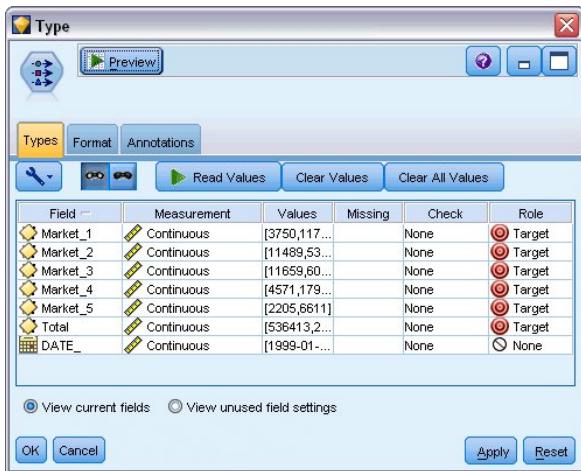


[Next](#)

Defining the Targets

1. Add a Type node and set the role to None for the *DATE_* field. Set the role to Target for all others (the *Market_n* fields plus the *Total* field).
2. Click the Read Values button to populate the Values column.

Figure 1. Setting the role for multiple fields

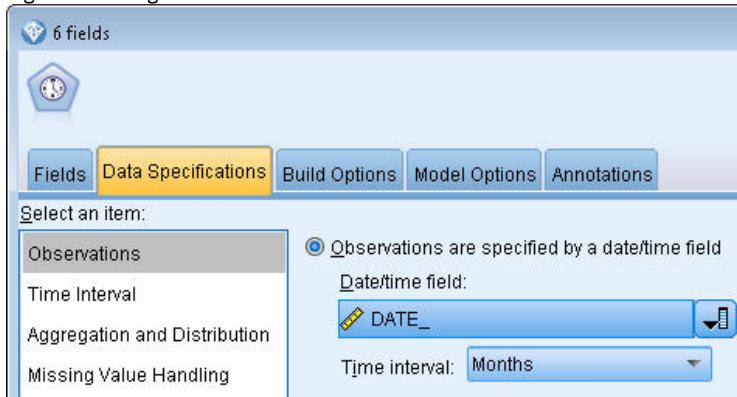


[Next](#)

Setting the Time Intervals

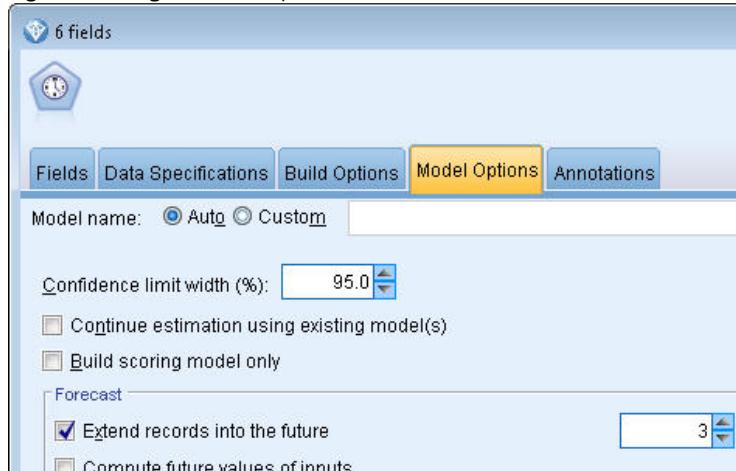
1. From the Modeling palette, add a Time Series node to the stream and attach it to the Type node.
2. On the Data Specifications tab, in the Observations pane, select **DATE_** as the Date/time field.
3. Select **Months** as the Time interval.

Figure 1. Setting the time interval



4. On the Model Options tab, select the Extend records into the future check box.
5. Set the value to 3.

Figure 2. Setting the forecast period

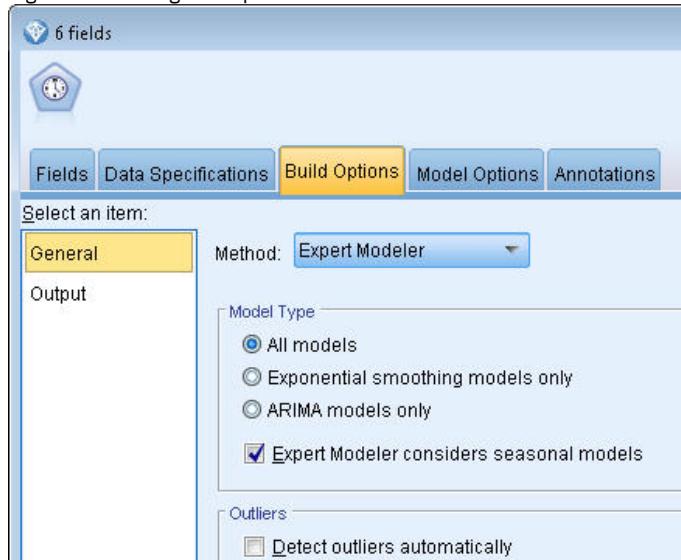


[Next](#)

Creating the Model

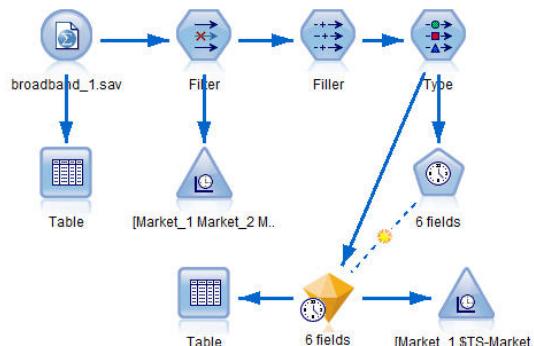
1. On the Time Series node, choose the Fields tab. In the Fields list, select all 5 of the markets and copy them to both the Targets and Candidate inputs lists. In addition, select and copy the Total field to the Targets list.
2. Choose the Build Options tab and, on the General pane, ensure the Expert Modeler Method is selected using all default settings. Doing so enables the Expert Modeler to decide the most appropriate model to use for each time series. Click Run.

Figure 1. Choosing the Expert Modeler for Time Series



3. Attach the Time Series model nugget to the Time Series node.
4. Attach a Table node to the Time Series model nugget and click Run.

Figure 2. Sample stream to show Time Series modeling



There are now three new rows (61 through 63) appended to the original data. These are the rows for the forecast period, in this case January to March 2004.

Several new columns are also present now; the `$TS-` columns are added by the Time Series node. The columns indicate the following for each row (that is, for each interval in the time series data):

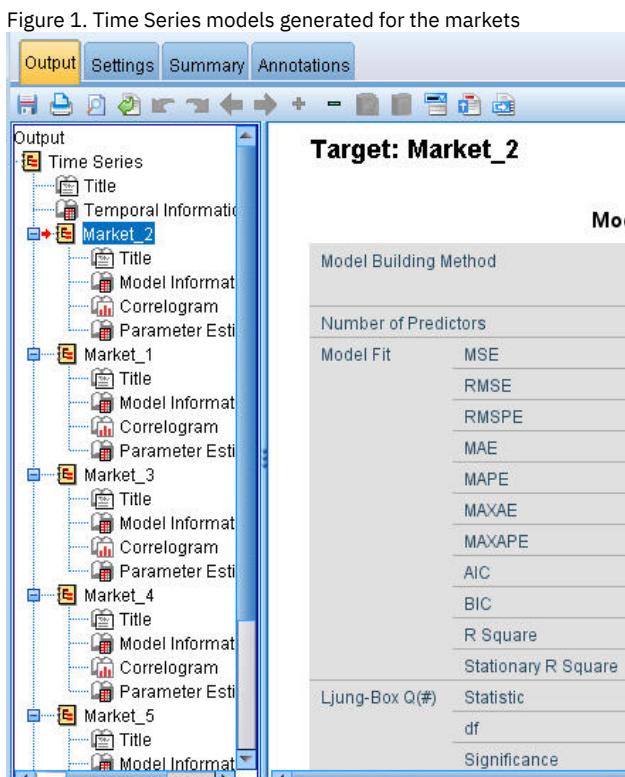
Column	Description
<code>\$TS-colname</code>	The generated model data for each column of the original data.
<code>\$TSLCI-colname</code>	The lower confidence interval value for each column of the generated model data.
<code>\$TSUCI-colname</code>	The upper confidence interval value for each column of the generated model data.
<code>\$TS-Total</code>	The total of the <code>\$TS-colname</code> values for this row.
<code>\$TSLCI-Total</code>	The total of the <code>\$TSLCI-colname</code> values for this row.
<code>\$TSUCI-Total</code>	The total of the <code>\$TSUCI-colname</code> values for this row.

The most significant columns for the forecast operation are the `$TS-Market_n`, `$TSLCI-Market_n`, and `$TSUCI-Market_n` columns. In particular, these columns in rows 61 through 63 contain the user subscription forecast data and confidence intervals for each of the local markets.

[Next](#)

Examining the model

1. Double-click the Time Series model nugget, and select the Output tab to display data about the models generated for each of the markets.



In the left Output column, select the Model Information for any of the Markets. The Number of Predictors line shows how many fields were used as predictors for each target; in this case, none.

The remaining lines in the Model Information tables show various goodness-of-fit measures for each model. The Stationary R Square value provides an estimate of the proportion of the total variation in the series that is explained by the model. The higher the value (to a maximum of 1.0), the better the fit of the model.

The Q(#) Statistic, df, and Significance lines relate to the Ljung-Box statistic, a test of the randomness of the residual errors in the model; the more random the errors, the better the model is likely to be. Q(#) is the Ljung-Box statistic itself, while df (degrees of freedom) indicates the number of model parameters that are free to vary when estimating a particular target.

The Significance line gives the significance value of the Ljung-Box statistic, providing another indication of whether the model is correctly specified. A significance value less than 0.05 indicates that the residual errors are not random, implying that there is structure in the observed series that is not accounted for by the model.

Taking both the Stationary R Square and Significance values into account, the models that the Expert Modeler has chosen for *Market_3*, and *Market_4* are quite acceptable. The Significance values for *Market_1*, *Market_2*, and *Market_5* are all less than 0.05, indicating that some experimentation with better-fitting models for these markets might be necessary.

The display shows a number of additional goodness-of-fit measures. The R Square value gives an estimation of the total variation in the time series that can be explained by the model. As the maximum value for this statistic is 1.0, our models are fine in this respect.

RMSE is the root mean square error, a measure of how much the actual values of a series differ from the values predicted by the model, and is expressed in the same units as those used for the series itself. As this is a measurement of an error, we want this value to be as low as possible. At first sight it appears that the models for *Market_2* and *Market_3*, while still acceptable according to the statistics we have seen so far, are less successful than those for the other three markets.

These additional goodness-of-fit measures include the mean absolute percentage errors (MAPE) and its maximum value (MAXAPE). Absolute percentage error is a measure of how much a target series varies from its model-predicted level, expressed as a percentage value. By examining the mean and maximum across all models, you can get an indication of the uncertainty in your predictions.

The MAPE value shows that all models display a mean uncertainty of around 1%, which is very low. The MAXAPE value displays the maximum absolute percentage error and is useful for imagining a worst-case scenario for your forecasts. It shows that the largest percentage error for most of the models falls in the range of roughly 1.8 to 3.7%, again a very low set of figures, with only *Market_4* being higher at nearer 7%.

The MAE (mean absolute error) value shows the mean of the absolute values of the forecast errors. Like the RMSE value, this is expressed in the same units as those used for the series itself. MAXAE shows the largest forecast error in the same units and indicates worst-case scenario for the forecasts.

Interesting though these absolute values are, it is the values of the percentage errors (MAPE and MAXAPE) that are more useful in this case, as the target series represent subscriber numbers for markets of varying sizes.

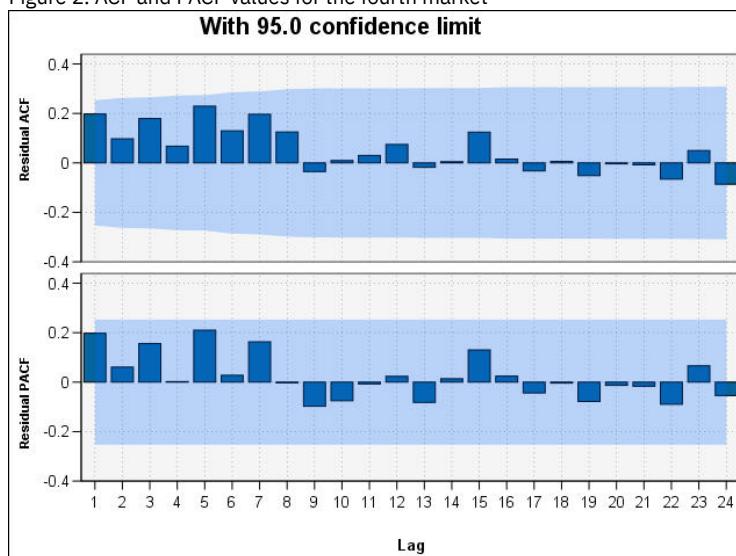
Do the MAPE and MAXAPE values represent an acceptable amount of uncertainty with the models? They are certainly very low. This is a situation in which business sense comes into play, because acceptable risk will change from problem to problem. We'll assume that the goodness-of-fit statistics fall within acceptable bounds and go on to look at the residual errors.

Examining the values of the autocorrelation function (ACF) and partial autocorrelation function (PACF) for the model residuals provides more quantitative insight into the models than simply viewing goodness-of-fit statistics.

A well-specified time series model will capture all of the nonrandom variation, including seasonality, trend, and cyclic and other factors that are important. If this is the case, any error should not be correlated with itself (autocorrelated) over time. A significant structure in either of the autocorrelation functions would imply that the underlying model is incomplete.

2. For the fourth market, in the left column, click Correlogram to display the values of the autocorrelation function (ACF) and partial autocorrelation function (PACF) for the residual errors in the model.

Figure 2. ACF and PACF values for the fourth market



In these plots, the original values of the error variable have been lagged by up to 24 time periods and compared with the original value to see if there is any correlation over time. For the model to be acceptable, none of the bars in the upper (ACF) plot should extend outside the shaded area, in either a positive (up) or negative (down) direction.

Should this occur, you would need to check the lower (PACF) plot to see whether the structure is confirmed there. The PACF plot looks at correlations after controlling for the series values at the intervening time points.

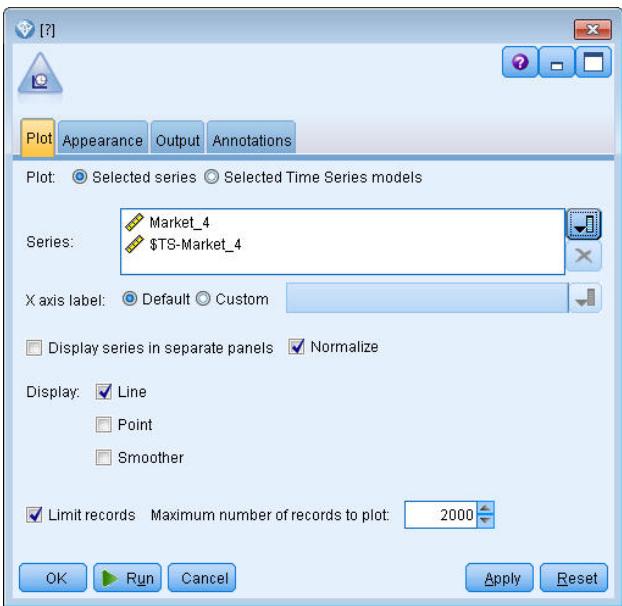
The values for *Market_4* are all within the shaded area, so we can continue and check the values for the other markets.

3. Click the Correlogram for each of the other markets and the totals.

The values for the other markets all show some values outside the shaded area, confirming what we suspected earlier from their Significance values. We'll need to experiment with some different models for those markets at some point to see if we can get a better fit, but for the rest of this example, we'll concentrate on what else we can learn from the *Market_4* model.

4. From the Graphs palette, attach a Time Plot node to the Time Series model nugget.
5. On the Plot tab, clear the Display series in separate panels check box.
6. At the Series list, click the field selector button, select the *Market_4* and *\$TS-Market_4* fields, and click OK to add them to the list.
7. Click Run to display a line graph of the actual and forecast data for the first of the local markets.

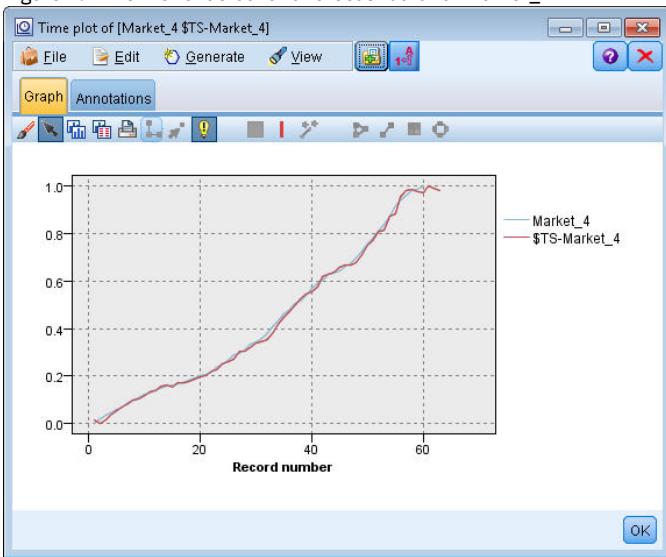
Figure 3. Selecting the fields to plot



Notice how the forecast (\$TS-Market_4) line extends past the end of the actual data. You now have a forecast of expected demand for the next three months in this market.

The lines for actual and forecast data over the entire time series are very close together on the graph, indicating that this is a reliable model for this particular time series.

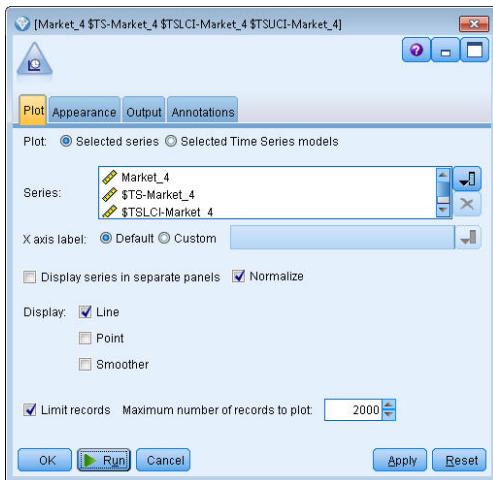
Figure 4. Time Plot of actual and forecast data for Market_4



Save the model in a file for use in a future example:

8. Click OK to close the current graph.
 9. Open the Time Series model nugget.
 10. Choose File > Save Node and specify the file location.
 11. Click Save.
- You have a reliable model for this particular market, but what margin of error does the forecast have? You can get an indication of this by examining the confidence interval.
12. Double-click the last Time Plot node in the stream (the one labeled Market_4 \$TS-Market_4) to open its dialog box again.
 13. Click the field selector button and add the \$TSLCI-Market_4 and \$TSUCI-Market_4 fields to the Series list.
 14. Click Run.

Figure 5. Adding more fields to plot

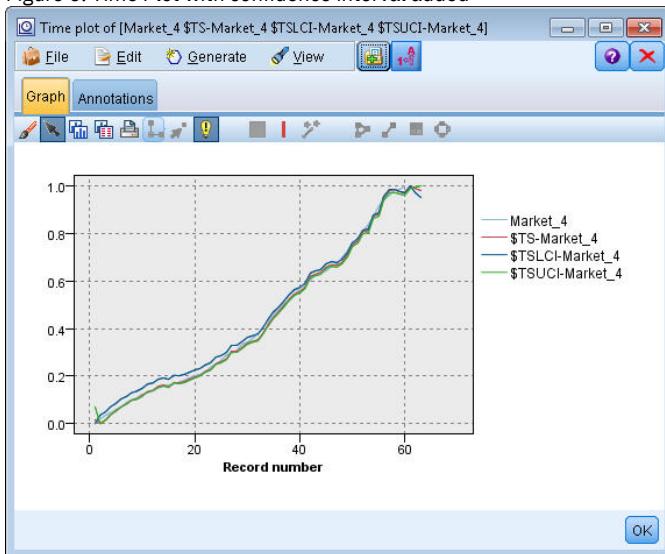


Now you have the same graph as before, but with the upper ($\$TSLCI$) and lower ($\$TSLCI$) limits of the confidence interval added.

Notice how the boundaries of the confidence interval diverge over the forecast period, indicating increasing uncertainty as you forecast further into the future.

However, as each time period goes by, you will have another (in this case) month's worth of actual usage data on which to base your forecast. You can read the new data into the stream and reapply your model now that you know it is reliable. See the topic [Reapplying a Time Series Model](#) for more information.

Figure 6. Time Plot with confidence interval added



[Next](#)

Summary

You have learned how to use the Expert Modeler to produce forecasts for multiple time series, and you have saved the resulting models to an external file.

In the next example, you will see how to transform nonstandard time series data into a format suitable for input to a Time Series node.

[Next](#)

Reapplying a Time Series Model

This example applies the time series models from the first time series example but can also be used independently. See the topic [Forecasting with the Time Series Node](#) for more information.

As in the original scenario, an analyst for a national broadband provider is required to produce monthly forecasts of user subscriptions for each of a number of local markets, in order to predict bandwidth requirements. You have already used the Expert Modeler to create models and to forecast three months into the future.

Your data warehouse has now been updated with the actual data for the original forecast period, so you would like to use that data to extend the forecast horizon by another three months.

This example uses the stream named *broadband_apply_models.str*, which references the data file named *broadband_2.sav*. These files are available from the *Demos* folder of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *broadband_apply_models.str* file is in the *streams* folder.

[Next](#)

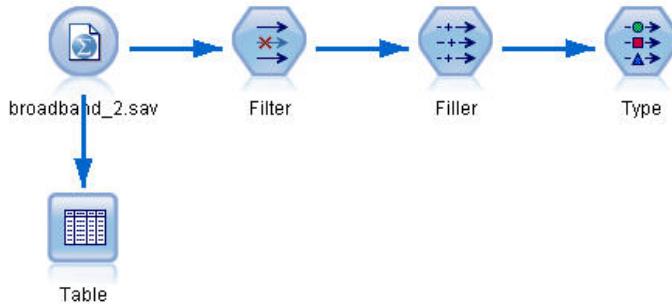
- [Retrieving the Stream](#)
- [Retrieving the saved model](#)
- [Generating a Modeling Node](#)
- [Generating a New Model](#)
- [Examining the New Model](#)
- [Summary](#)

Retrieving the Stream

In this example, you'll be recreating a Time Series node from the Time Series model saved in the first example. Don't worry if you don't have a model saved, we've provided one in the *Demos* folder.

1. Open the stream *broadband_apply_models.str* from the *streams* folder under *Demos*.

Figure 1. Opening the stream



The updated monthly data is collected in *broadband_2.sav*.

2. Attach a Table node to the IBM® SPSS® Statistics File source node, open the Table node and click Run.

Note: The data file has been updated with the actual sales data for January through March 2004, in rows 61 to 63.

Figure 2. Updated sales data

A screenshot of the SPSS Modeler Table node interface. The title bar says "Table (89 fields, 63 records)". The main area shows a data grid with columns labeled: #1, Market_82, Market_83, Market_84, Market_85, Total, YEAR_, MONTH_, DATE_. The data starts at row 44 and continues to row 63. The first few rows of data are:

#1	Market_82	Market_83	Market_84	Market_85	Total	YEAR_	MONTH_	DATE_
44	58620	20482	14326	16935	17917...	2002	8	AUG 2002
45	60119	21211	14349	17179	18249...	2002	9	SEP 2002
46	61320	21893	14333	17601	18601...	2002	10	OCT 2002
47	63099	22471	14223	17816	18945...	2002	11	NOV 2002
48	64687	23112	14514	17937	19343...	2002	12	DEC 2002
49	65518	23686	14856	18003	19752...	2003	1	JAN 2003
50	65570	24669	15182	17875	20148...	2003	2	FEB 2003
51	66567	25469	15709	18214	20540...	2003	3	MAR 2003
52	67527	25868	16155	18557	20922...	2003	4	APR 2003
53	67724	26284	16521	19190	21300...	2003	5	MAY 2003
54	68644	26468	16567	19938	21669...	2003	6	JUN 2003
55	69878	26781	16618	20876	22004...	2003	7	JUL 2003
56	71538	27566	16553	21514	22398...	2003	8	AUG 2003
57	73162	28164	16597	21779	22773...	2003	9	SEP 2003
58	74167	28693	16669	22266	23160...	2003	10	OCT 2003
59	76036	28922	16748	22559	23616...	2003	11	NOV 2003
60	76630	29811	16798	23018	24067...	2003	12	DEC 2003
61	79002	30034	17122	23160	24509...	2004	1	JAN 2004
62	81123	30091	17581	23698	24968...	2004	2	FEB 2004
63	83909	30162	17894	24355	25383...	2004	3	MAR 2004

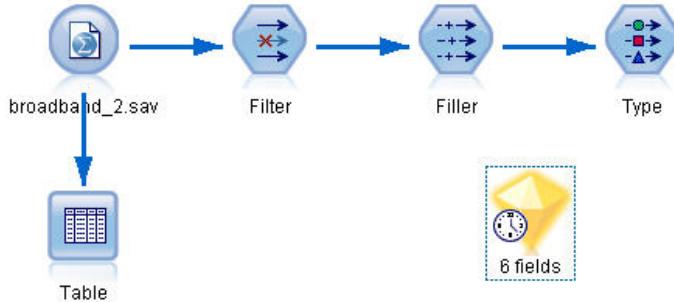
[Next](#)

Retrieving the saved model

1. On the IBM® SPSS® Modeler menu, choose Insert > Node From File and select the *TSmodel.nod* file from the *Demos* folder (or use the Time Series model you saved in the first time series example).

This file contains the time series models from the previous example. The insert operation places the corresponding Time Series model nugget on the canvas.

Figure 1. Adding the model nugget



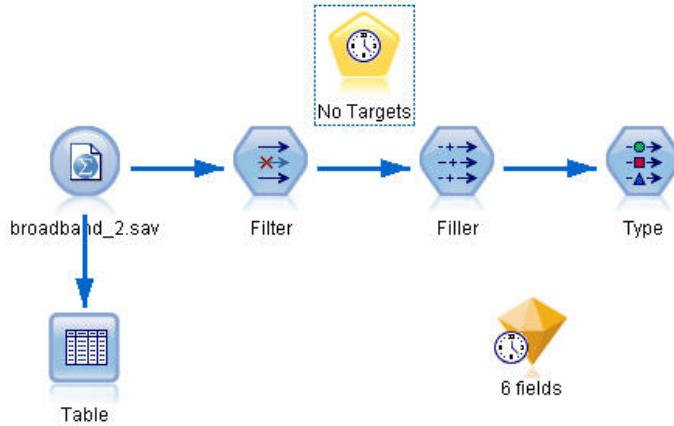
[Next](#)

Generating a Modeling Node

1. Open the Time Series model nugget and choose Generate > Generate Modeling Node.

This places a Time Series modeling node on the canvas.

Figure 1. Generating a modeling node from the model nugget



[Next](#)

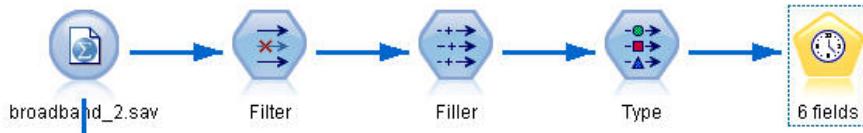
Generating a New Model

1. Close the Time Series model nugget and delete it from the canvas.

The old model was built on 60 rows of data. You need to generate a new model based on the updated sales data (63 rows).

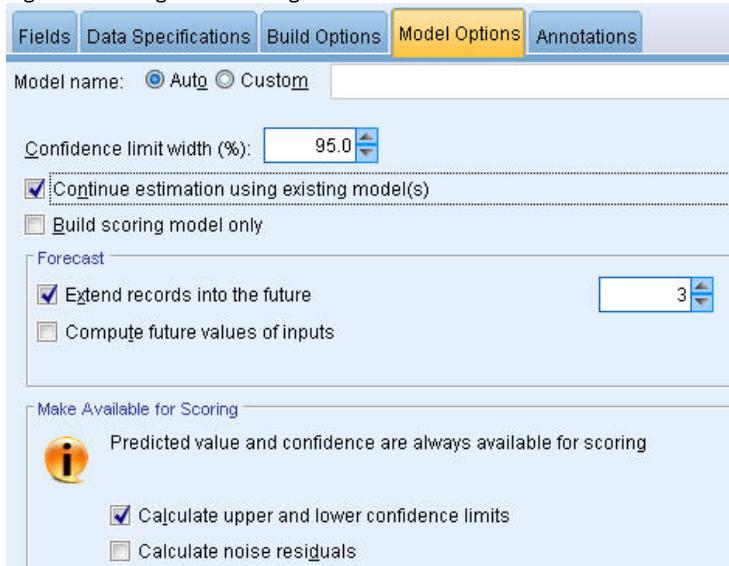
2. Attach the newly generated Time Series build node to the stream.

Figure 1. Attaching the modeling node to the stream



3. Open the Time Series node.
4. On the Model Options tab, ensure that Continue estimation using existing models is checked.

Figure 2. Reusing stored settings for the time series model



5. Ensure that Extend records into the future is set to 3.
6. Click Run to place a new model nugget on the canvas and in the Models palette.

[Next](#)

Examining the New Model

1. Attach a Table node to the new Time Series model nugget on the canvas.
2. Open the Table node and click Run.

The new model still forecasts three months ahead because you're reusing the stored settings. However, this time it forecasts April through June (on lines 64 to 66) because the estimation period now ends in March instead of January.

Figure 1. Table showing new forecast

Table (26 fields, 66 records)

File Edit Generate

Table Annotations

	\$TS-Market_4	\$TSLCI-Market_4	\$TSUCI-Market_4	\$TS-Total	\$TSLCI-Total	\$TSL
47	13460.165	13046.567	13883.520	1895694.552	1890768.484	190
48	13637.234	13218.196	14066.159	1929821.249	1924806.501	193
49	14038.478	13607.110	14480.023	1974007.314	1968877.747	197
50	14588.176	14139.917	15047.010	2017063.960	2011822.507	202
51	14826.444	14370.864	15292.773	2055709.852	2050367.976	206
52	15328.900	14857.881	15811.032	2094273.974	2088831.887	209
53	15403.883	14930.559	15888.373	2131431.902	2125893.258	213
54	16187.796	15690.385	16696.942	2168729.836	2163094.271	217
55	16303.304	15802.343	16816.083	2204919.579	2199189.973	221
56	17250.576	16720.508	17793.149	2235223.381	2229415.030	224
57	17616.290	17074.985	18170.366	2278910.104	2272988.230	228
58	17639.270	17097.259	18194.069	2316079.288	2310060.827	232
59	17552.150	17012.816	18104.209	2355228.381	2349108.190	236
60	17499.120	16981.415	18049.510	2406836.211	2400581.914	241
61	18183.056	17624.336	18754.958	2453038.341	2446663.985	245
62	18512.777	17943.925	19095.050	2496354.087	2489867.172	250
63	19125.395	18537.719	19726.936	2543477.283	2536867.916	255
64	19394.782	18798.828	20004.796	2581510.338	2574802.140	258
65	19387.631	18551.891	20251.298	2625230.895	2611195.788	263
66	19550.898	18525.803	20617.962	2669744.972	2646565.409	269

OK

3. Attach a Time Plot graph node to the Time Series model nugget.

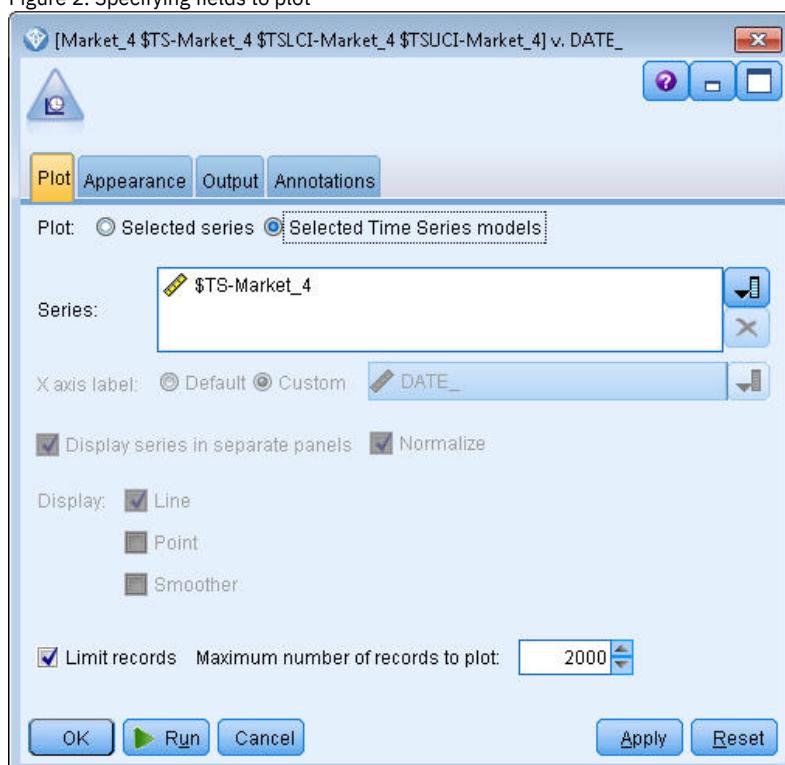
This time we'll use the time plot display designed especially for time series models.

4. On the Plot tab, set the X axis label to Custom, and select **Date_**.

5. For the Plot, choose the Selected Time Series models option.

6. From the Series list, click the field selector button, select the **\$TS-Market_4** field, and click OK to add it to the list.

Figure 2. Specifying fields to plot

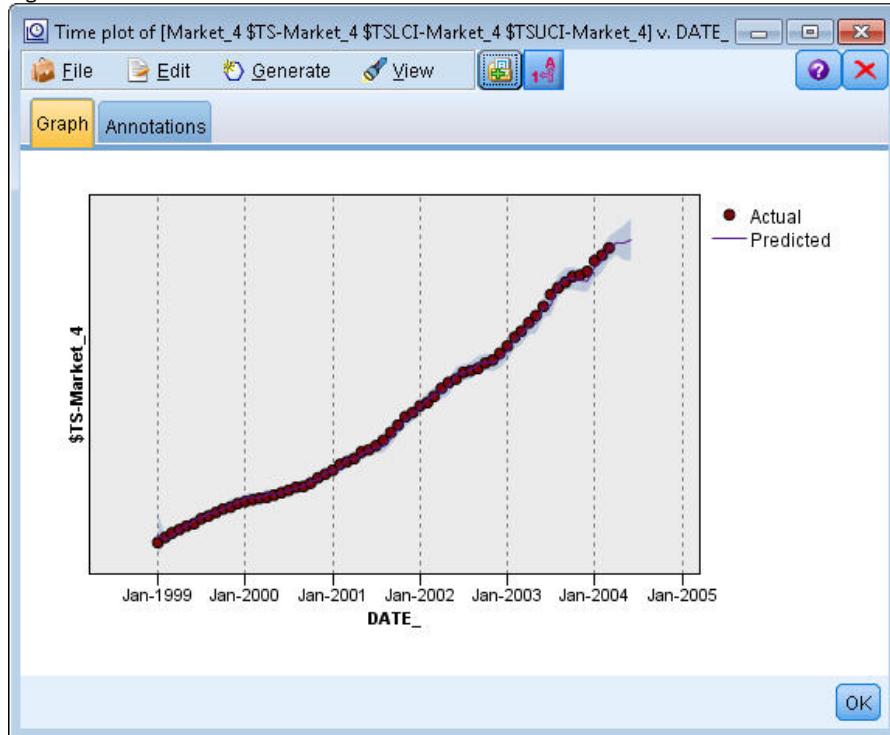


7. Click Run.

Now you have a graph that shows the actual sales for Market_4 up to March 2004, together with the forecast (Predicted) sales and the confidence interval (indicated by the blue shaded area) up to June 2004.

As in the first example, the forecast values follow the actual data closely throughout the time period, indicating once again that you have a good model.

Figure 3. Forecast extended to June



[Next](#)

Summary

You have learned how to apply saved models to extend your previous forecasts when more current data becomes available, and you have done this without rebuilding your models. Of course, if there is reason to think that a model has changed, you should rebuild it.

Forecasting Catalog Sales (Time Series)

A catalog company is interested in forecasting monthly sales of its men's clothing line, based on their sales data for the last 10 years.

This example uses the stream named *catalog_forecast.str*, which references the data file named *catalog_seasfac.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *catalog_forecast.str* file is in the *streams* directory.

We've seen in an earlier example how you can let the Expert Modeler decide which is the most appropriate model for your time series. Now it's time to take a closer look at the two methods that are available when choosing a model yourself--exponential smoothing and ARIMA.

To help you decide on an appropriate model, it's a good idea to plot the time series first. Visual inspection of a time series can often be a powerful guide in helping you choose. In particular, you need to ask yourself:

- Does the series have an overall trend? If so, does the trend appear constant or does it appear to be dying out with time?
- Does the series show seasonality? If so, do the seasonal fluctuations seem to grow with time or do they appear constant over successive periods?

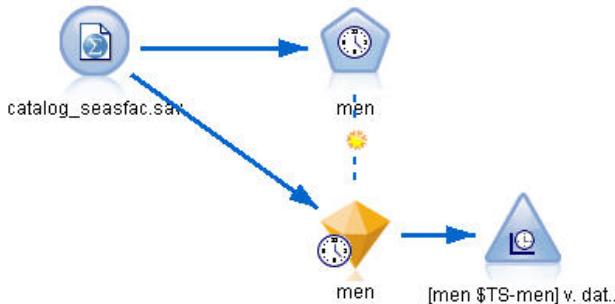
[Next](#)

- [Creating the Stream](#)
- [Examining the Data](#)
- [Exponential Smoothing](#)
- [ARIMA](#)
- [Summary](#)

Creating the Stream

1. Create a new stream and add a Statistics File source node pointing to *catalog_seasfac.sav*.

Figure 1. Forecasting catalog sales



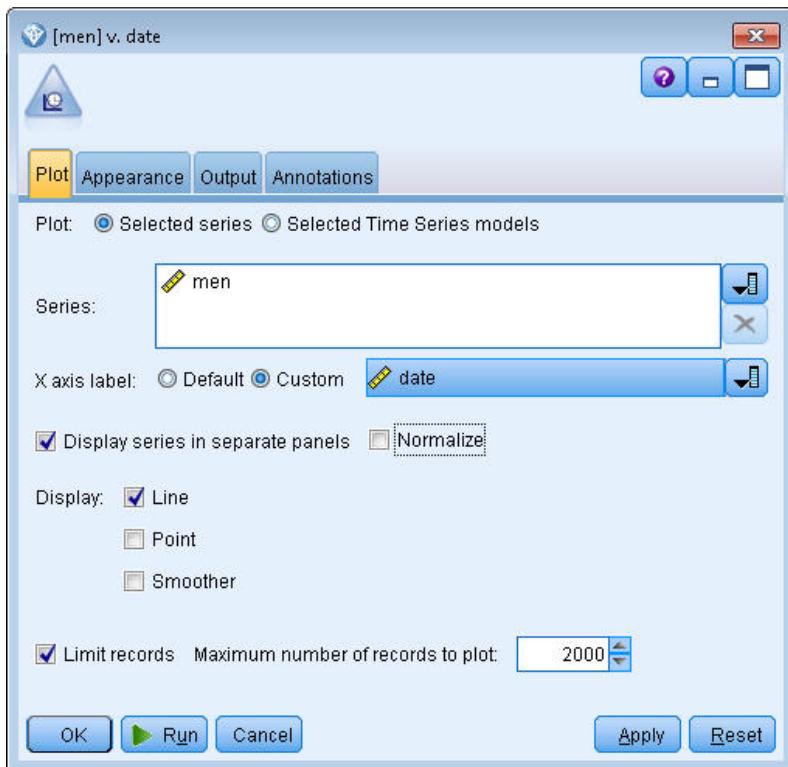
2. Open the IBM® SPSS® Statistics File source node and select the Types tab.
3. Click Read Values, then OK.
4. Click the Role column for the **men** field and set the role to Target.

Figure 2. Specifying the target field



5. Set the role for all the other fields to None, and click OK.
6. Attach a Time Plot graph node to the IBM SPSS Statistics File source node.
7. Open the Time Plot node and, on the Plot tab, add **men** to the Series list.
8. Set the X axis label to Custom, and select **date**.
9. Clear the Normalize check box.

Figure 3. Plotting the time series

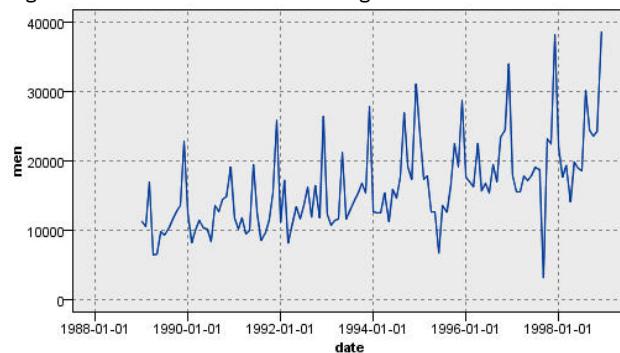


10. Click Run.

[Next](#)

Examining the Data

Figure 1. Actual sales of men's clothing



The series shows a general upward trend; that is, the series values tend to increase over time. The upward trend is seemingly constant, which indicates a linear trend.

The series also has a distinct seasonal pattern with annual highs in December, as indicated by the vertical lines on the graph. The seasonal variations appear to grow with the upward series trend, which suggests multiplicative rather than additive seasonality.

1. Click OK to close the plot.

Now that you've identified the characteristics of the series, you're ready to try modeling it. The exponential smoothing method is useful for forecasting series that exhibit trend, seasonality, or both. As we've seen, your data exhibit both characteristics.

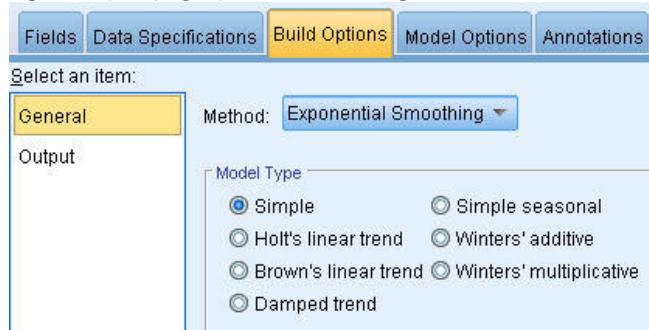
[Next](#)

Exponential Smoothing

Building a best-fit exponential smoothing model involves determining the model type (whether the model needs to include trend, seasonality, or both) and then obtaining the best-fit parameters for the chosen model.

The plot of men's clothing sales over time suggested a model with both a linear trend component and a multiplicative seasonality component. This implies a Winters' model. First, however, we will explore a simple model (no trend and no seasonality) and then a Holt's model (incorporates linear trend but no seasonality). This will give you practice in identifying when a model is not a good fit to the data, an essential skill in successful model building.

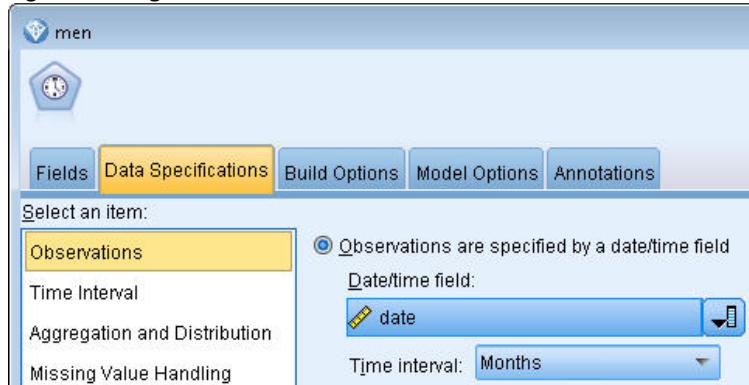
Figure 1. Specifying exponential smoothing



We'll start with a simple exponential smoothing model.

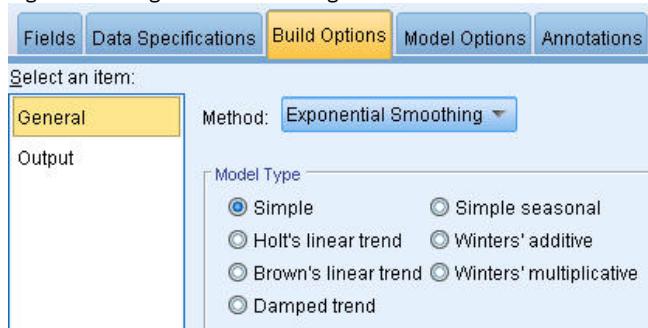
1. Add a Time Series node to the stream and attach it to the source node.
2. On the Data Specifications tab, in the Observations pane, select date as the Date/time field.
3. Select Months as the Time interval.

Figure 2. Setting the time interval



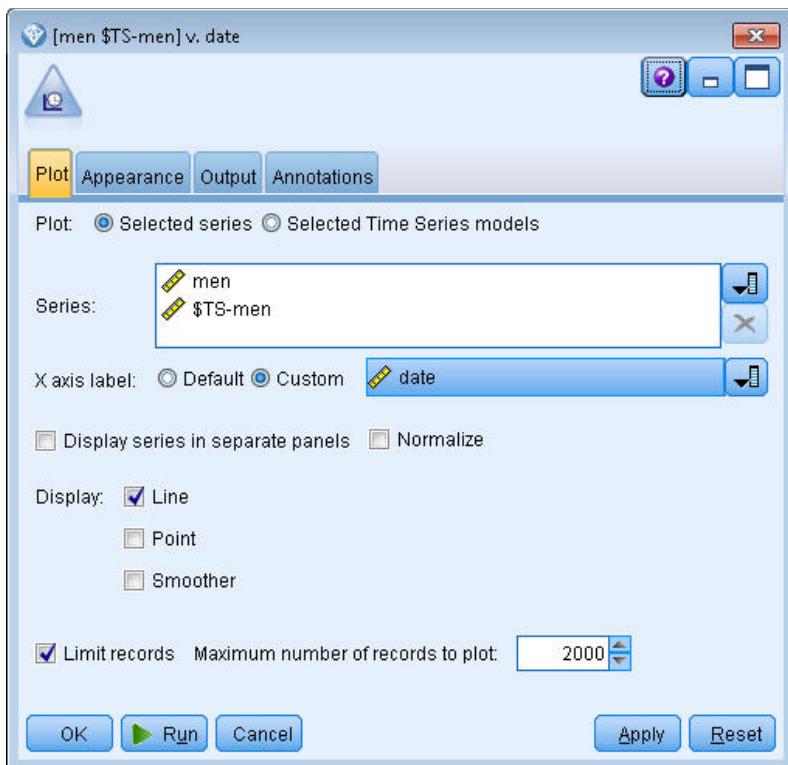
4. On the Build Options tab, in the General pane, set Method to Exponential Smoothing.
5. Set Model Type to Simple.

Figure 3. Setting the model building method



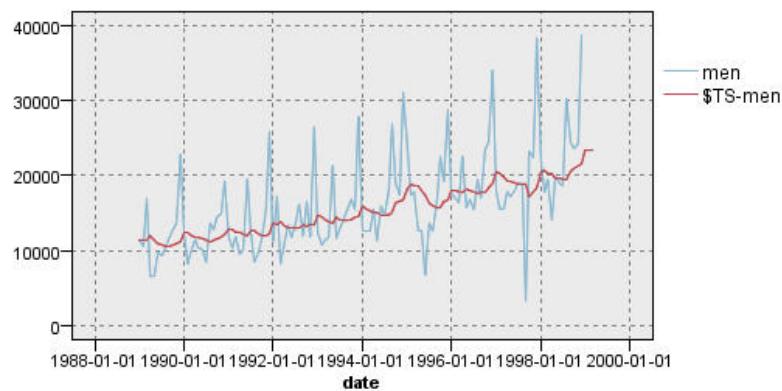
6. Click Run to create the model nugget.

Figure 4. Plotting the Time Series model



7. Attach a Time Plot node to the model nugget.
8. On the Plot tab, add men and \$TS-men to the Series list.
9. Set the X axis label to Custom, and select date.
10. Clear the Display series in separate panels and Normalize check boxes.
11. Click Run.

Figure 5. Simple exponential smoothing model

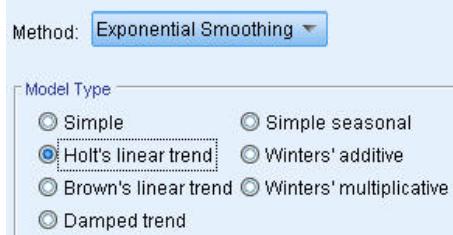


The men plot represents the actual data, while \$TS-men denotes the time series model.

Although the simple model does, in fact, exhibit a gradual (and rather ponderous) upward trend, it takes no account of seasonality. You can safely reject this model.

12. Click OK to close the time plot window.

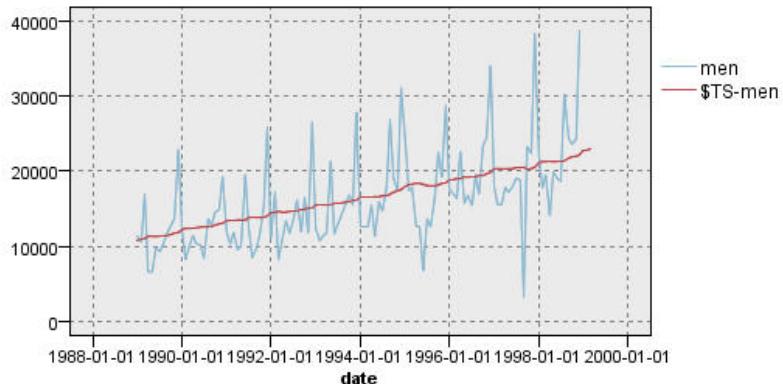
Figure 6. Selecting Holt's model



Let's try Holt's linear model. This should at least model the trend better than the simple model, although it too is unlikely to capture the seasonality.

13. Reopen the Time Series node.
14. On the Build Options tab, in the General pane, with Exponential Smoothing still selected as the Method, select Holts linear trend as the Model Type.
15. Click Run to re-create the model nugget.
16. Re-open the Time Plot node and click Run.

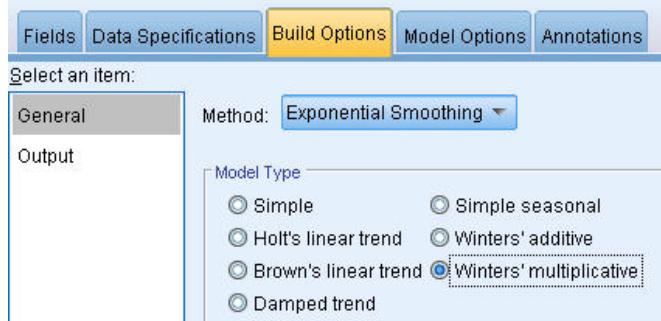
Figure 7. Holt's linear trend model



Holt's model displays a smoother upward trend than the simple model but it still takes no account of the seasonality, so you can discard this one too.

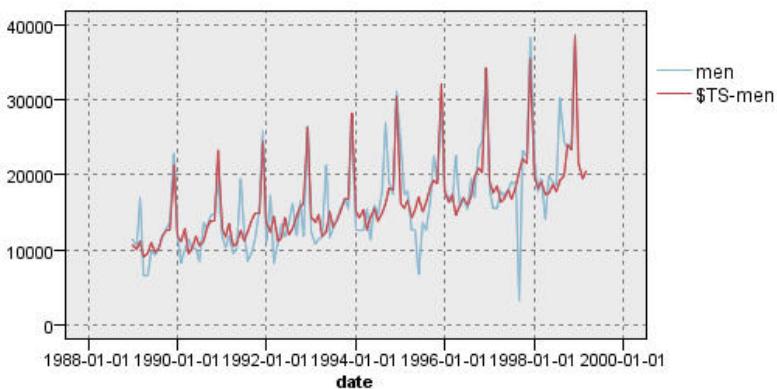
17. Close the time plot window.
- You may recall that the initial plot of men's clothing sales over time suggested a model incorporating a linear trend and multiplicative seasonality. A more suitable candidate, therefore, might be Winters' model.

Figure 8. Selecting Winters' model



18. Reopen the Time Series node.
19. On the Build Options tab, in the General pane, with Exponential Smoothing still selected as the Method, select Winters' multiplicative as the Model Type.
20. Click Run to re-create the model nugget.
21. Open the Time Plot node and click Run.

Figure 9. Winters' multiplicative model



This looks better; the model reflects both the trend and the seasonality of the data.

The dataset covers a period of 10 years and includes 10 seasonal peaks occurring in December of each year. The 10 peaks present in the predicted results match up well with the 10 annual peaks in the real data.

However, the results also underscore the limitations of the Exponential Smoothing procedure. Looking at both the upward and downward spikes, there is significant structure that is not accounted for.

If you are primarily interested in modeling a long-term trend with seasonal variation, then exponential smoothing may be a good choice. To model a more complex structure such as this one, we need to consider using the ARIMA procedure.

[Next](#)

ARIMA

With the ARIMA procedure you can create an autoregressive integrated moving-average (ARIMA) model that is suitable for finely tuned modeling of time series. ARIMA models provide more sophisticated methods for modeling trend and seasonal components than do exponential smoothing models, and they have the added benefit of being able to include predictor variables in the model.

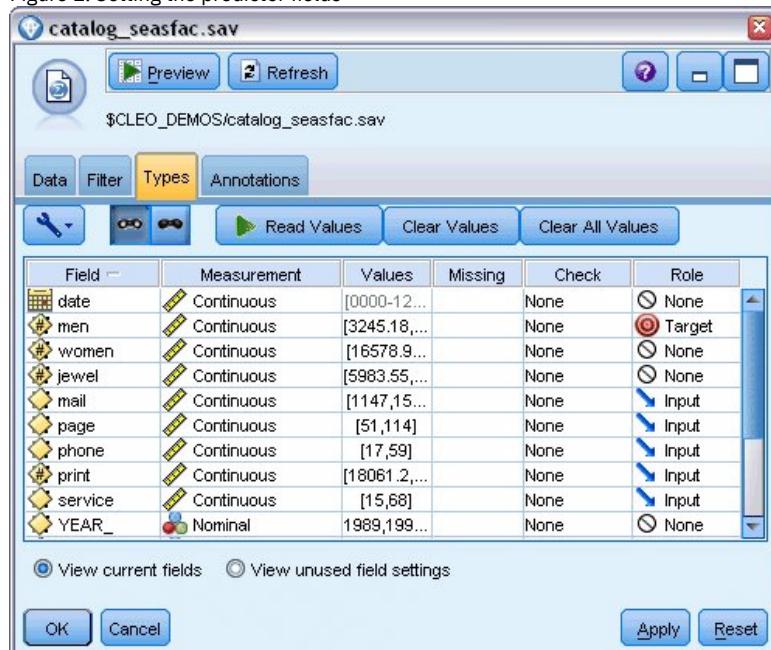
Continuing the example of the catalog company that wants to develop a forecasting model, we have seen how the company has collected data on monthly sales of men's clothing along with several series that might be used to explain some of the variation in sales. Possible predictors include the number of catalogs mailed and the number of pages in the catalog, the number of phone lines open for ordering, the amount spent on print advertising, and the number of customer service representatives.

Are any of these predictors useful for forecasting? Is a model with predictors really better than one without? Using the ARIMA procedure, we can create a forecasting model with predictors, and see if there is a significant difference in predictive ability over the exponential smoothing model with no predictors.

With the ARIMA method, you can fine-tune the model by specifying orders of autoregression, differencing, and moving average, as well as seasonal counterparts to these components. Determining the best values for these components manually can be a time-consuming process involving a good deal of trial and error so, for this example, we'll let the Expert Modeler choose an ARIMA model for us.

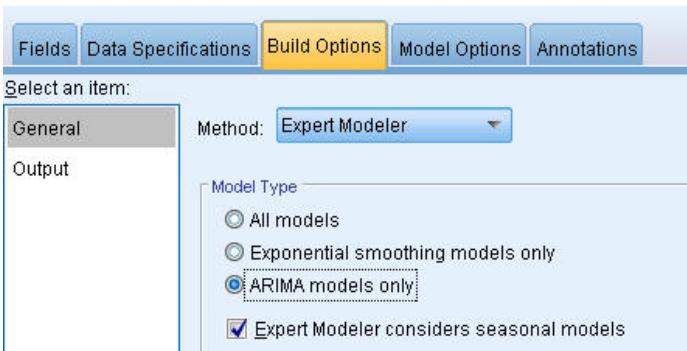
We'll try to build a better model by treating some of the other variables in the dataset as predictor variables. The ones that seem most useful to include as predictors are the number of catalogs mailed (`mail`), the number of pages in the catalog (`page`), the number of phone lines open for ordering (`phone`), the amount spent on print advertising (`print`), and the number of customer service representatives (`service`).

Figure 1. Setting the predictor fields



1. Open the IBM® SPSS® Statistics File source node.
2. On the Types tab, set the Role for `mail`, `page`, `phone`, `print`, and `service` to Input.
3. Ensure that the role for `men` is set to Target and that all the remaining fields are set to None.
4. Click OK.
5. Open the Time Series node.
6. On the Build Options tab, in the General pane, set Method to Expert Modeler.
7. Select the ARIMA models only option and ensure that Expert Modeler considers seasonal models is checked.

Figure 2. Choosing only ARIMA models

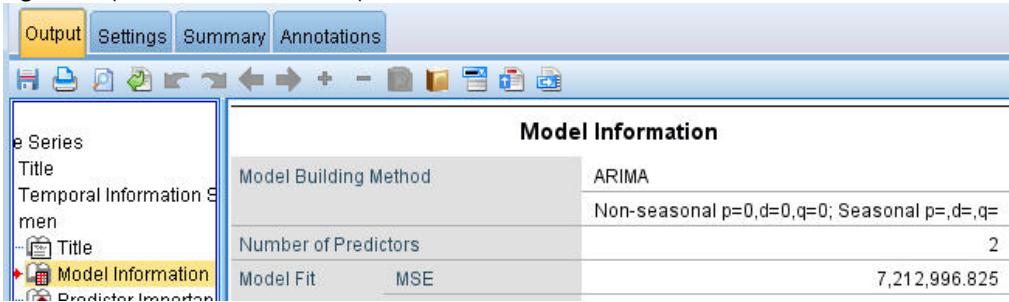


8. Click Run to re-create the model nugget.

9. Open the model nugget.

On the Output tab, in the left column, select the Model information. Notice how the Expert Modeler has chosen only two of the five specified predictors as being significant to the model.

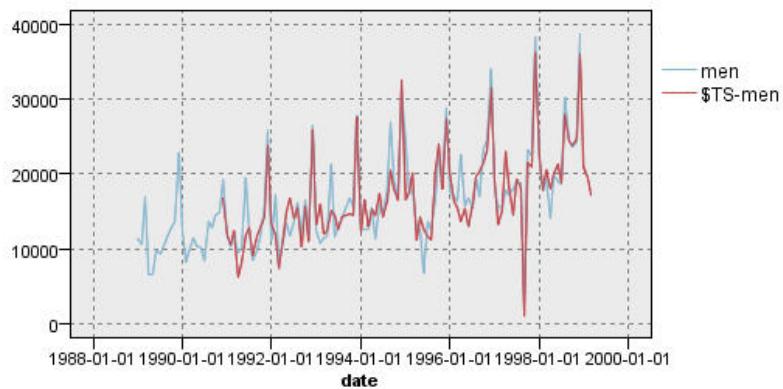
Figure 3. Expert Modeler chooses two predictors



10. Click OK to close the model nugget.

11. Open the Time Plot node and click Run.

Figure 4. ARIMA model with predictors specified



This model improves on the previous one by capturing the large downward spike as well, making it the best fit so far.

We could try refining the model even further, but any improvements from this point on are likely to be minimal. We've established that the ARIMA model with predictors is preferable, so let's use the model we have just built. For the purposes of this example, we'll forecast sales for the coming year.

12. Click OK to close the time plot window.

13. Open the Time Series node and select the Model Options tab.

14. Select the Extend records into the future checkbox and set its value to 12.

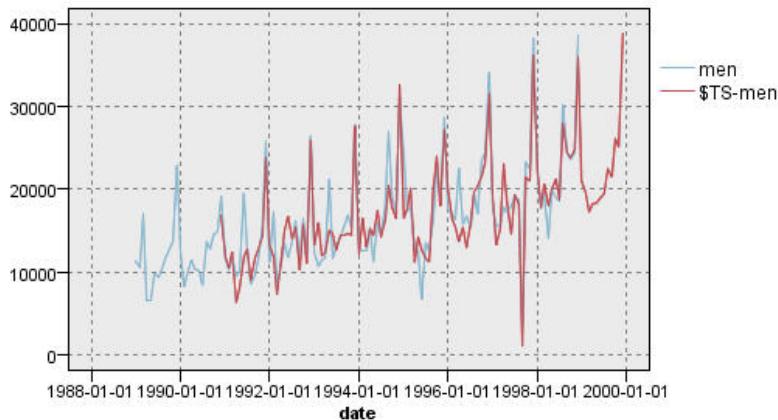
15. Select the Compute future values of inputs checkbox.

16. Click Run to re-create the model nugget.

17. Open the Time Plot node and click Run.

The forecast for 1999 looks good; as expected, there's a return to normal sales levels following the December peak, and a steady upward trend in the second half of the year, with sales in general above those for the previous year.

Figure 5. Sales forecast extended by 12 months



[Next](#)

Summary

You have successfully modeled a complex time series, incorporating not only an upward trend but also seasonal and other variations. You have also seen how, through trial and error, you can get closer and closer to an accurate model, which you have then used to forecast future sales.

In practice, you would need to reapply the model as your actual sales data are updated--for example, every month or every quarter--and produce updated forecasts. See the topic [Reapplying a Time Series Model](#) for more information.

Making Offers to Customers (Self-Learning)

The Self-Learning Response Model (SLRM) node generates and enables the updating of a model that allows you to predict which offers are most appropriate for customers and the probability of the offers being accepted. These sorts of models are most beneficial in customer relationship management, such as marketing applications or call centers.

This example is based on a fictional banking company. The marketing department wants to achieve more profitable results in future campaigns by matching the right offer of financial services to each customer. Specifically, the example uses a Self-Learning Response Model to identify the characteristics of customers who are most likely to respond favorably based on previous offers and responses and to promote the best current offer based on the results.

This example uses the stream *pm_selflearn.str*, which references the data files *pm_customer_train1.sav*, *pm_customer_train2.sav*, and *pm_customer_train3.sav*. These files are available from the *Demos* folder of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *pm_selflearn.str* file is in the *streams* folder.

Existing Data

The company has historical data tracking the offers made to customers in past campaigns, along with the responses to those offers. These data also include demographic and financial information that can be used to predict response rates for different customers.

Figure 1. Responses to previous offers

Table (31 fields, 21,927 records)

	customer_id	campaign	response	response_date	purchase	purchase_date	product_id	Rowid
1	7	2	0	\$null\$	0	\$null\$	\$null\$	1
2	13	2	0	\$null\$	0	\$null\$	\$null\$	2
3	15	2	0	\$null\$	0	\$null\$	\$null\$	3
4	16	2	1	2006-07-05 00:00:00	0	\$null\$	183	761
5	23	2	0	\$null\$	0	\$null\$	\$null\$	4
6	24	2	0	\$null\$	0	\$null\$	\$null\$	5
7	30	2	0	\$null\$	0	\$null\$	\$null\$	6
8	30	3	0	\$null\$	0	\$null\$	\$null\$	7
9	33	2	0	\$null\$	0	\$null\$	\$null\$	8
10	42	3	0	\$null\$	0	\$null\$	\$null\$	9
11	42	2	0	\$null\$	0	\$null\$	\$null\$	10
12	52	2	0	\$null\$	0	\$null\$	\$null\$	11
13	57	2	0	\$null\$	0	\$null\$	\$null\$	12
14	63	2	1	2006-07-14 00:00:00	0	\$null\$	183	1501
15	74	2	0	\$null\$	0	\$null\$	\$null\$	13
16	74	3	0	\$null\$	0	\$null\$	\$null\$	14
17	75	2	0	\$null\$	0	\$null\$	\$null\$	15
18	82	2	0	\$null\$	0	\$null\$	\$null\$	16
19	89	3	0	\$null\$	0	\$null\$	\$null\$	17
20	89	2	0	\$null\$	0	\$null\$	\$null\$	18

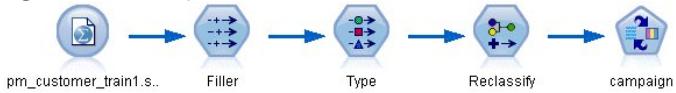
[Next](#)

- [Building the Stream](#)
- [Browsing the model](#)

Building the Stream

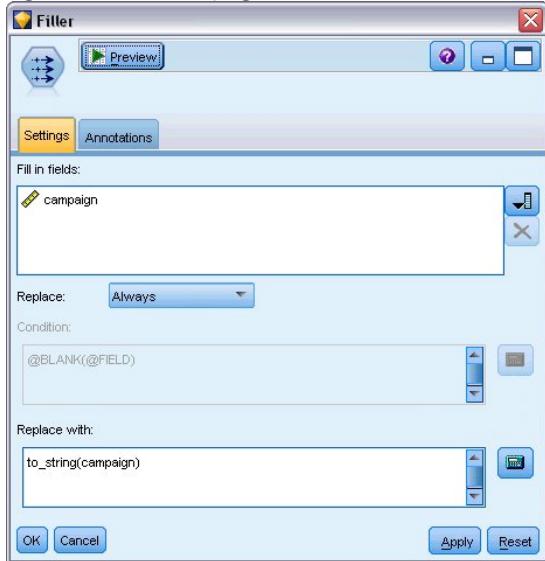
1. Add a Statistics File source node pointing to *pm_customer_train1.sav*, located in the *Demos* folder of your IBM® SPSS® Modeler installation.

Figure 1. SLM sample stream



2. Add a Filler node and select *campaign* as the Fill in field.
3. Select a Replace type of Always.
4. In the Replace with text box, enter *to_string(campaign)* and click OK.

Figure 2. Derive a campaign field



5. Add a Type node, and set the Role to None for the *customer_id*, *response_date*, *purchase_date*, *product_id*, *Rowid*, and *X_random* fields.

Figure 3. Changing the Type node settings



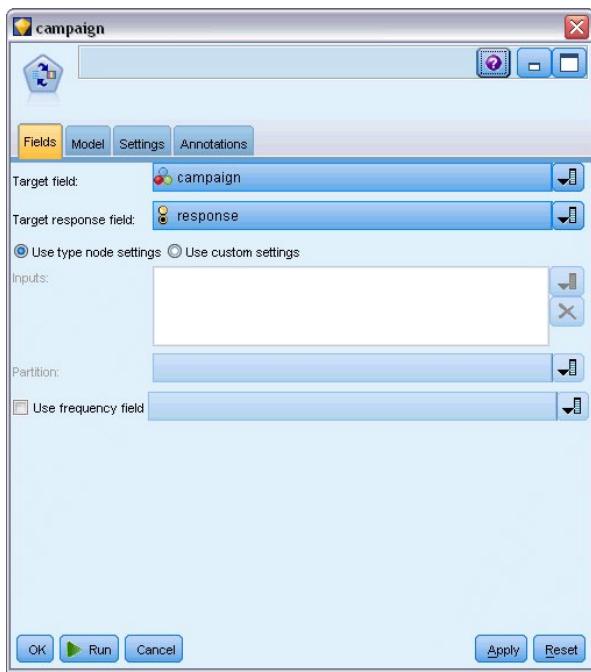
6. Set the *Role* to Target for the *campaign* and *response* fields. These are the fields on which you want to base your predictions.
Set the Measurement to Flag for the *response* field.
7. Click Read Values, then OK.
Because the campaign field data show as a list of numbers (1, 2, 3, and 4), you can reclassify the fields to have more meaningful titles.
8. Add a Reclassify node to the Type node.
9. In the Reclassify into field, select Existing field.
10. In the Reclassify field list, select campaign.
11. Click the Get button; the campaign values are added to the *Original value* column.
12. In the *New value* column, enter the following campaign names in the first four rows:
 - Mortgage
 - Car loan
 - Savings
 - Pension
13. Click OK.

Figure 4. Reclassify the campaign names



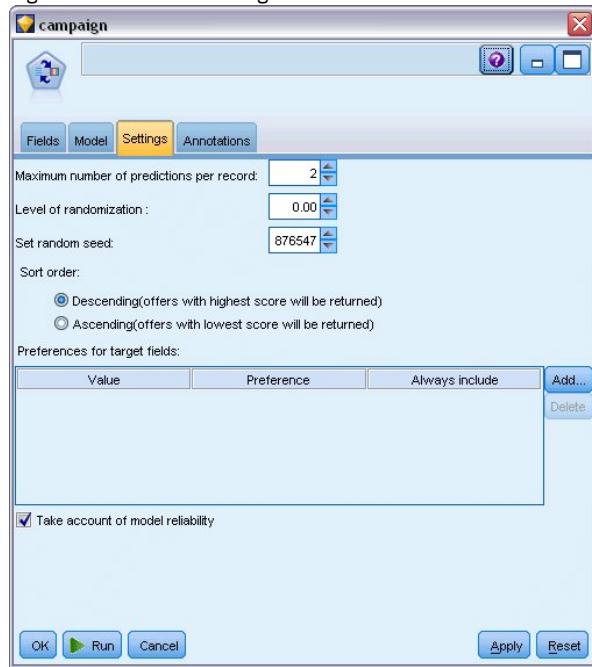
14. Attach an SLRM modeling node to the Reclassify node. On the Fields tab, select campaign for the Target field, and response for the Target response field.

Figure 5. Select the target and target response



15. On the Settings tab, in the Maximum number of predictions per record field, reduce the number to 2.
This means that for each customer, there will be two offers identified that have the highest probability of being accepted.
16. Ensure that Take account of model reliability is selected, and click Run.

Figure 6. SLM node settings



[Next](#)

Browsing the model

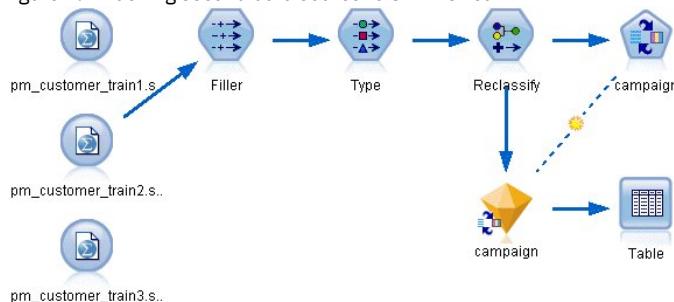
1. Open the model nugget. The Model tab initially shows the estimated the accuracy of the predictions for each offer and the relative importance of each predictor in estimating the model.
To display the correlation of each predictor with the target variable, choose Association with Response from the View list in the right-hand pane.
2. To switch between each of the four offers for which there are predictions, select the required offer from the View list in the left-hand pane.

Figure 1. SLM model nugget



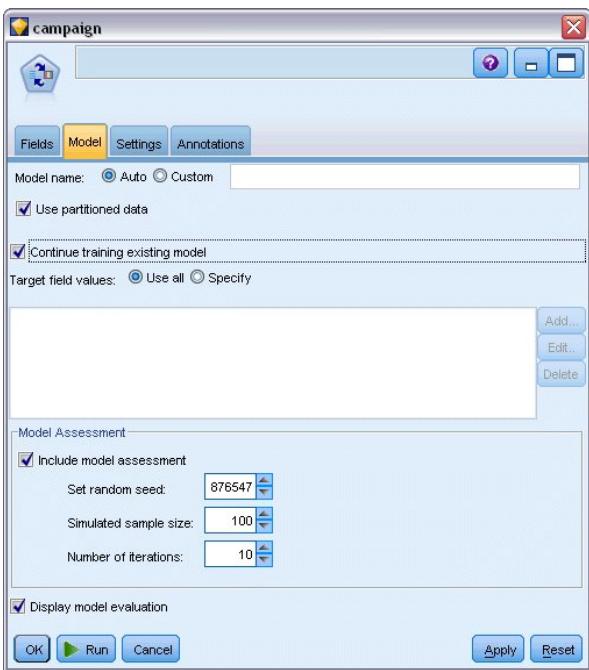
3. Close the model nugget window.
4. On the stream canvas, disconnect the IBM® SPSS® Statistics File source node pointing to *pm_customer_train1.sav*.
5. Add a Statistics File source node pointing to *pm_customer_train2.sav*, located in the *Demos* folder of your IBM SPSS Modeler installation, and connect it to the Filler node.

Figure 2. Attaching second data source to SLM stream



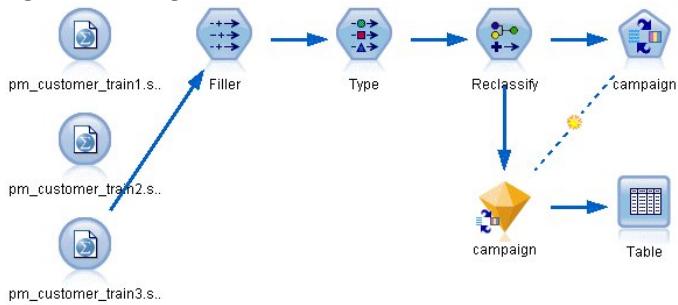
6. On the Model tab of the SLM node, select Continue training existing model.

Figure 3. Continue training model



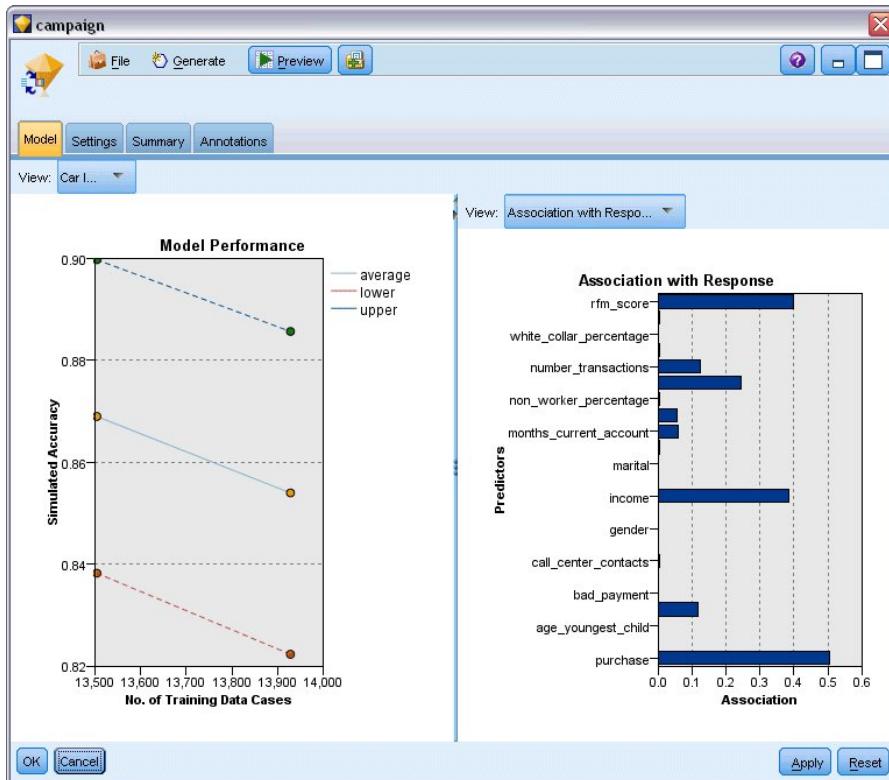
7. Click Run to re-create the model nugget. To view its details, double-click the nugget on the canvas. The Model tab now shows the revised estimates of the accuracy of the predictions for each offer.
8. Add a Statistics File source node pointing to *pm_customer_train3.sav*, located in the *Demos* folder of your IBM SPSS Modeler installation, and connect it to the Filler node.

Figure 4. Attaching third data source to SLM stream



9. Click Run to re-create the model nugget once more. To view its details, double-click the nugget on the canvas.
 10. The Model tab now shows the final estimated accuracy of the predictions for each offer.
- As you can see, the average accuracy fell slightly (from 86.9% to 85.4%) as you added the additional data sources; however, this fluctuation is a minimal amount and may be attributed to slight anomalies within the available data.

Figure 5. Updated SLM model nugget



11. Attach a Table node to the last (third) generated model and execute the Table node.
12. Scroll across to the right of the table. The predictions show which offers a customer is most likely to accept and the confidence that they will accept, depending on each customer's details.

For example, in the first line of the table shown, there is only a 13.2% confidence rating (denoted by the value 0.132 in the \$SC-campaign-1 column) that a customer who previously took out a car loan will accept a pension if offered one. However, the second and third lines show two more customers who also took out a car loan; in their cases, there is a 95.7% confidence that they, and other customers with similar histories, would open a savings account if offered one, and over 80% confidence that they would accept a pension.

Figure 6. Model output - predicted offers and confidences

Table (35 fields, 27 records)					
	X_random	\$S-campaign-1	\$SC-campaign-1	\$S-campaign-2	\$SC-campaign-2
1	1	Pension	0.132	Mortgage	0.107
2	1	Savings	0.957	Pension	0.844
3	1	Savings	0.957	Pension	0.802
4	3	Pension	0.132	Mortgage	0.107
5	1	Pension	0.805	Savings	0.284
6	3	Pension	0.132	Mortgage	0.107
7	2	Pension	0.132	Mortgage	0.107
8	3	Pension	0.132	Mortgage	0.107
9	1	Pension	0.132	Mortgage	0.107
10	1	Pension	0.132	Mortgage	0.107
11	2	Pension	0.132	Mortgage	0.107
12	2	Pension	0.132	Mortgage	0.107
13	2	Savings	0.957	Mortgage	0.829
14	2	Savings	0.164	Pension	0.132
15	2	Savings	0.957	Pension	0.868
16	2	Pension	0.132	Mortgage	0.107
17	3	Pension	0.132	Mortgage	0.107
18	3	Pension	0.132	Mortgage	0.107
19	3	Savings	0.289	Pension	0.132
20	2	Pension	0.132	Mortgage	0.107

Explanations of the mathematical foundations of the modeling methods used in IBM SPSS Modeler are listed in the *IBM SPSS Modeler Algorithms Guide*, available as a PDF file as part of your product download.

Note also that these results are based on the training data only. To assess how well the model generalizes to other data in the real world, you would use a Partition node to hold out a subset of records for purposes of testing and validation.

Predicting Loan Defaulters (Bayesian Network)

Bayesian networks enable you to build a probability model by combining observed and recorded evidence with "common-sense" real-world knowledge to establish the likelihood of occurrences by using seemingly unlinked attributes.

This example uses the stream named *bayes_bankloan.str*, which references the data file named *bankloan.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation and can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *bayes_bankloan.str* file is in the *streams* directory.

For example, suppose a bank is concerned about the potential for loans not to be repaid. If previous loan default data can be used to predict which potential customers are liable to have problems repaying loans, these "bad risk" customers can either be declined a loan or offered alternative products.

This example focuses on using existing loan default data to predict potential future defaulters, and looks at three different Bayesian network model types to establish which is better at predicting in this situation.

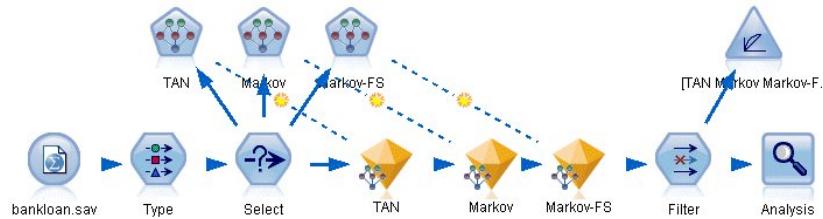
[Next](#)

- [Building the Stream](#)
- [Browsing the model](#)

Building the Stream

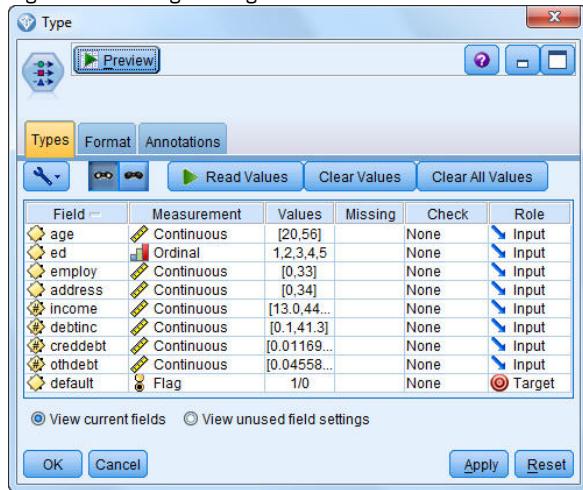
1. Add a Statistics File source node pointing to *bankloan.sav* in the *Demos* folder.

Figure 1. Bayesian Network sample stream



2. Add a Type node to the source node and set the role of the default field to Target. All other fields should have their role set to Input.
3. Click the Read Values button to populate the *Values* column.

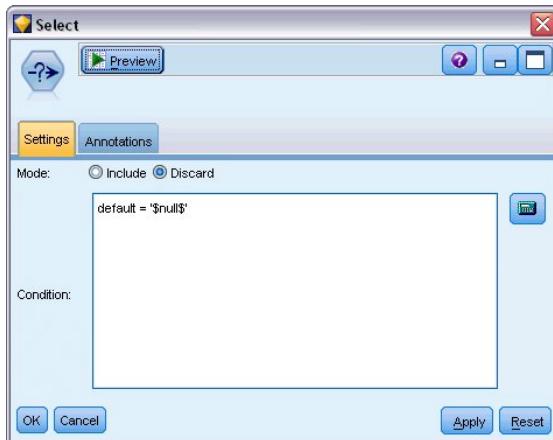
Figure 2. Selecting the target field



Cases where the target has a null value are of no use when building the model. You can exclude those cases to prevent them from being used in model evaluation.

4. Add a Select node to the Type node.
5. For Mode, select Discard.
6. In the Condition box, enter default = '\$null\$'.

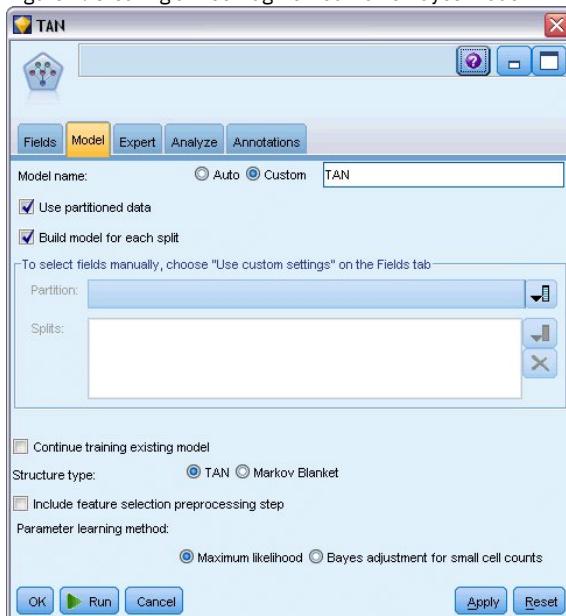
Figure 3. Discarding null targets



Because you can build several different types of Bayesian networks, it is worth comparing several to see which model provides the best predictions. The first one to create is a Tree Augmented Naïve Bayes (TAN) model.

7. Attach a Bayesian Network node to the Select node.
8. On the Model tab, for Model name, select Custom and enter TAN in the text box.
9. For Structure type, select TAN and click OK.

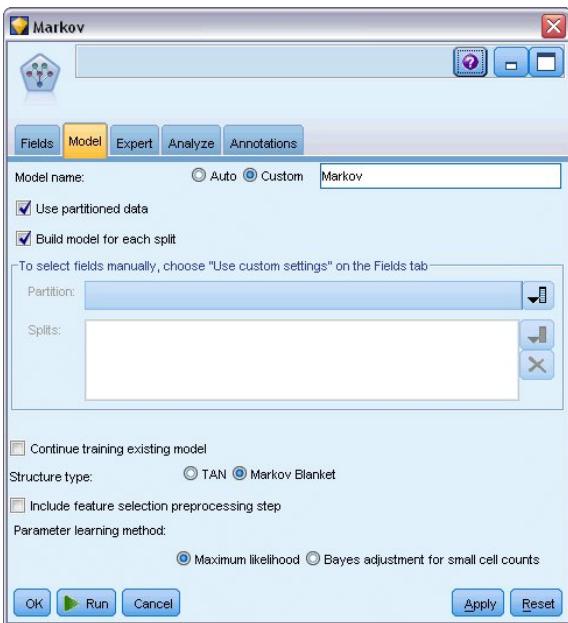
Figure 4. Creating a Tree Augmented Naïve Bayes model



The second model type to build has a Markov Blanket structure.

10. Attach a second Bayesian Network node to the Select node.
11. On the Model tab, for Model name, select Custom and enter Markov in the text box.
12. For Structure type, select Markov Blanket and click OK.

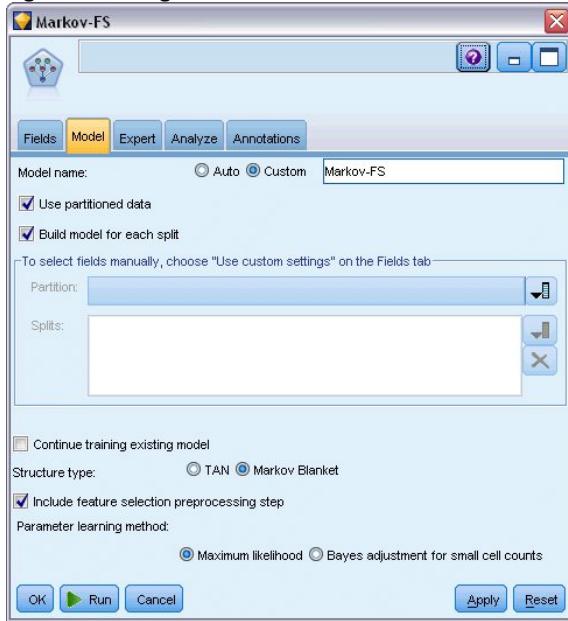
Figure 5. Creating a Markov Blanket model



The third model type to build has a Markov Blanket structure and also uses feature selection preprocessing to select the inputs that are significantly related to the target variable.

13. Attach a third Bayesian Network node to the Select node.
14. On the Model tab, for Model name, select Custom and enter Markov-FS in the text box.
15. For Structure type, select Markov Blanket.
16. Select Include feature selection preprocessing step and click OK.

Figure 6. Creating a Markov Blanket model with Feature Selection preprocessing



[Next](#)

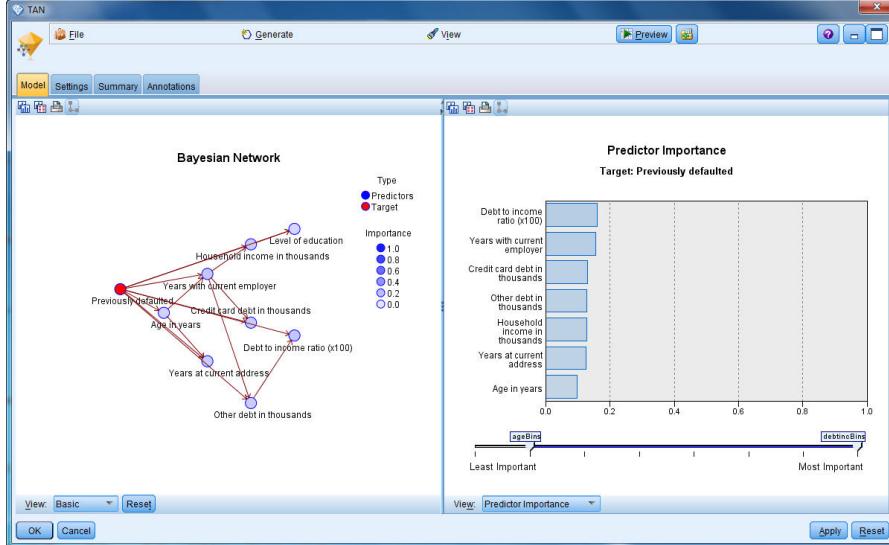
Browsing the model

1. Run the stream to create the model nuggets, which are added to the stream and to the Models palette in the upper-right corner. To view their details, double-click on any of the model nuggets in the stream.

The model nugget Model tab is split into two panes. The left pane contains a network graph of nodes that displays the relationship between the target and its most important predictors, as well as the relationship between the predictors.

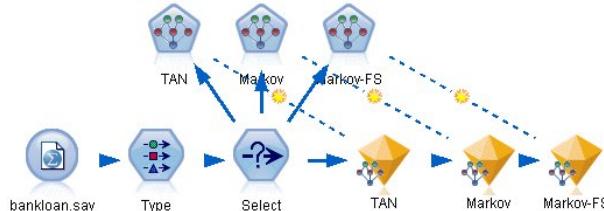
The right pane shows either *Predictor Importance*, which indicates the relative importance of each predictor in estimating the model, or *Conditional Probabilities*, which contains the conditional probability value for each node value and each combination of values in its parent nodes.

Figure 1. Viewing a Tree Augmented Naïve Bayes model



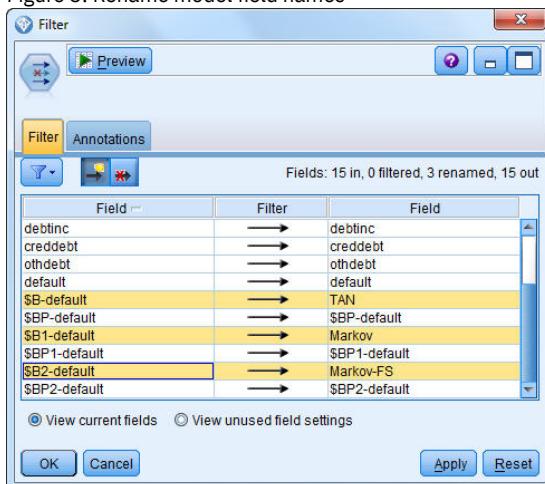
2. Connect the TAN model nugget to the Markov nugget (choose Replace on the warning dialog).
3. Connect the Markov nugget to the Markov-FS nugget (choose Replace on the warning dialog).
4. Align the three nuggets with the Select node for ease of viewing.

Figure 2. Aligning the nuggets in the stream



5. To rename the model outputs for clarity on the Evaluation graph that you'll be creating, attach a Filter node to the Markov-FS model nugget.
6. In the right Field column, rename \$B-default as TAN, \$B1-default as Markov, and \$B2-default as Markov-FS.

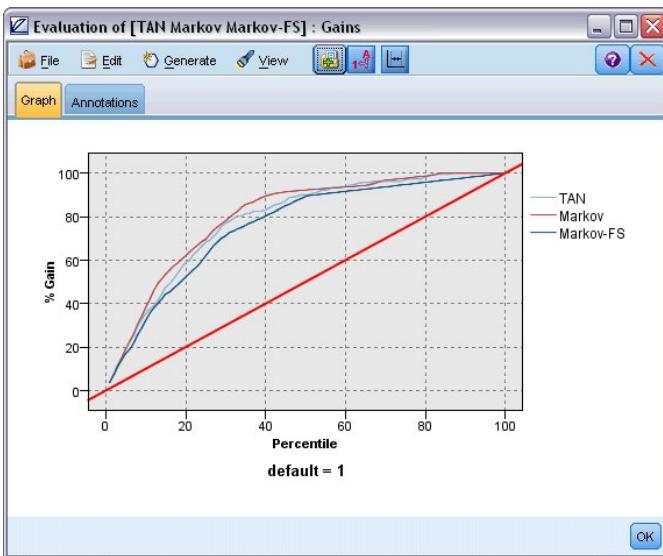
Figure 3. Rename model field names



To compare the models' predicted accuracy, you can build a gains chart.

7. Attach an Evaluation graph node to the Filter node and execute the graph node using its default settings. The graph shows that each model type produces similar results; however, the Markov model is slightly better.

Figure 4. Evaluating model accuracy

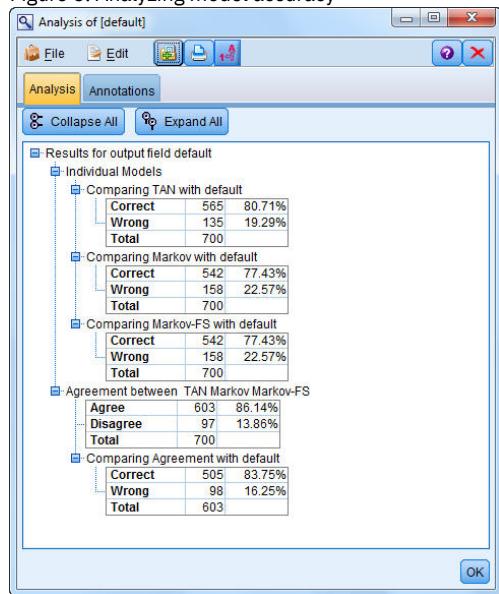


To check how well each model predicts, you could use an Analysis node instead of the Evaluation graph. This shows the accuracy in terms of percentage for both correct and incorrect predictions.

8. Attach an Analysis node to the Filter node and execute the Analysis node using its default settings.

As with the Evaluation graph, this shows that the Markov model is slightly better at predicting correctly; however, the Markov-FS model is only a few percentage points behind the Markov model. This may mean it would be better to use the Markov-FS model since it uses fewer inputs to calculate its results, thereby saving on data collection and entry time and processing time.

Figure 5. Analyzing model accuracy



Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the *IBM SPSS Modeler Algorithms Guide*, available from the *|Documentation* directory of the installation disk.

Note also that these results are based on the training data only. To assess how well the model generalizes to other data in the real world, you would use a Partition node to hold out a subset of records for purposes of testing and validation.

Retraining a Model on a Monthly Basis (Bayesian Network)

Bayesian networks enable you to build a probability model by combining observed and recorded evidence with "common-sense" real-world knowledge to establish the likelihood of occurrences by using seemingly unlinked attributes.

This example uses the stream named *bayes_churn_retrain.str*, which references the data files named *telco_Jan.sav* and *telco_Feb.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation and can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *bayes_churn_retrain.str* file is in the *streams* directory.

For example, suppose that a telecommunications provider is concerned about the number of customers it is losing to competitors (churn). If historic customer data can be used to predict which customers are more likely to churn in the future, these customers can be targeted with incentives or other offers to discourage them from transferring to another service provider.

This example focuses on using an existing month's churn data to predict which customers may be likely to churn in the future and then adding the following month's data to refine and retrain the model.

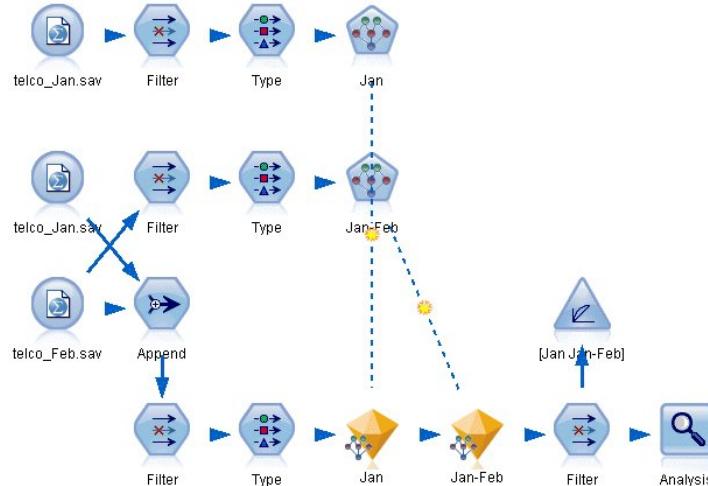
[Next](#)

- [Building the Stream](#)
- [Evaluating the model](#)

Building the Stream

1. Add a Statistics File source node pointing to *telco_Jan.sav* in the *Demos* folder.

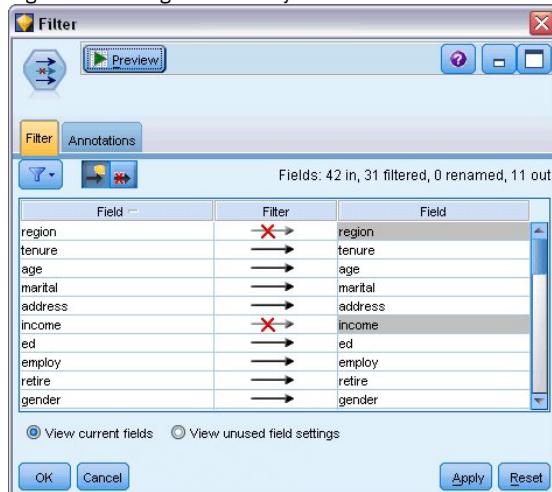
Figure 1. Bayesian Network sample stream



Previous analysis has shown you that several data fields are of little importance when predicting churn. These fields can be filtered from your data set to increase the speed of processing when you are building and scoring models.

2. Add a Filter node to the Source node.
3. Exclude all fields except *address*, *age*, *churn*, *custcat*, *ed*, *employ*, *gender*, *marital*, *reside*, *retire*, and *tenure*.
4. Click OK.

Figure 2. Filtering unnecessary fields



5. Add a Type node to the Filter node.
6. Open the Type node and click the Read Values button to populate the *Values* column.
7. In order that the Evaluation node can assess which value is true and which is false, set the measurement level for the *churn* field to Flag, and set its role to Target. Click OK.

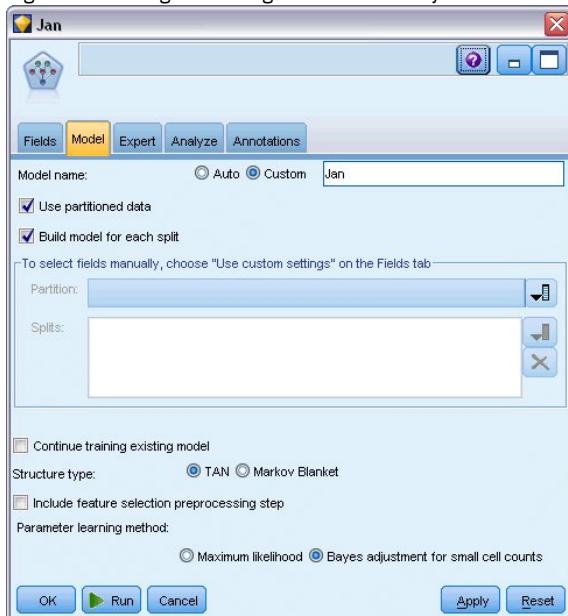
Figure 3. Selecting the target field



You can build several different types of Bayesian networks; however, for this example you are going to build a Tree Augmented Naïve Bayes (TAN) model. This creates a large network and ensures that you have included all possible links between data variables, thereby building a robust initial model.

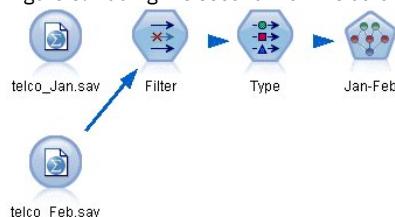
8. Attach a Bayesian Network node to the Type node.
9. On the Model tab, for Model name, select Custom and enter Jan in the text box.
10. For Parameter learning method, select Bayes adjustment for small cell counts.
11. Click Run. The model nugget is added to the stream, and also to the Models palette in the upper-right corner.

Figure 4. Creating a Tree Augmented Naïve Bayes model



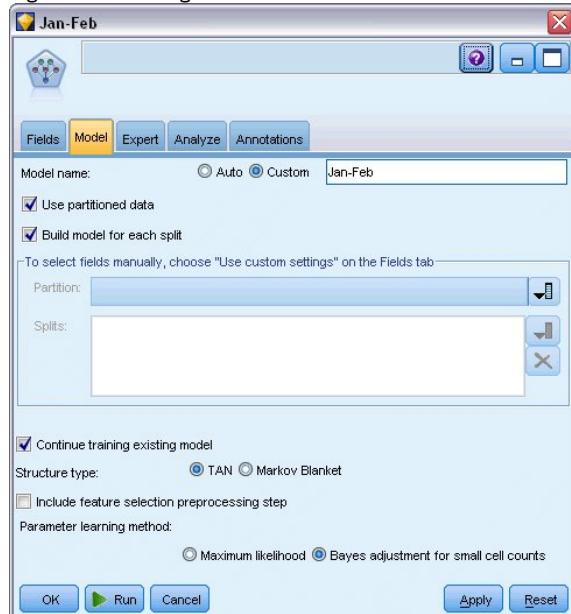
12. Add a Statistics File source node pointing to telco_Feb.sav in the Demos folder.
13. Attach this new source node to the Filter node (on the warning dialog, choose Replace to replace the connection to the previous source node).

Figure 5. Adding the second month's data



14. On the Model tab of the Bayesian Network node, for Model name, select Custom and enter Jan-Feb in the text box.
15. Select Continue training existing model.
16. Click Run. The model nugget overwrites the existing one in the stream, but is also added to the Models palette in the upper-right corner.

Figure 6. Retraining the model



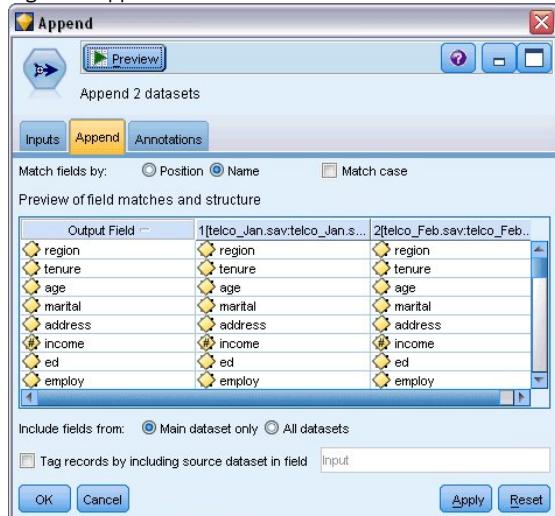
[Next](#)

Evaluating the model

To compare the models, you must combine the two datasets.

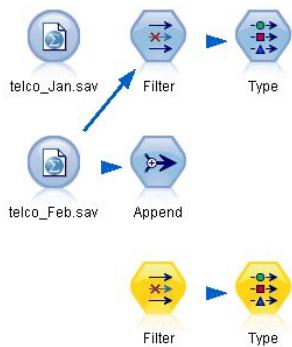
1. Add an Append node and attach both the *telco_Jan.sav* and *telco_Feb.sav* source nodes to it.

Figure 1. Append the two data sources



2. Copy the Filter and Type nodes from earlier in the stream and paste them onto the stream canvas.
3. Attach the Append node to the newly copied Filter node.

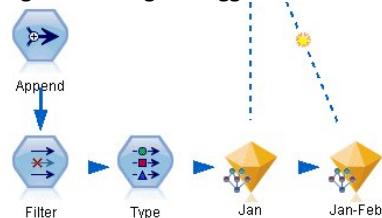
Figure 2. Pasting the copied nodes into the stream



The nuggets for the two Bayesian Network models are located in the Models palette in the upper-right corner.

4. Double-click the Jan model nugget to bring it into the stream, and attach it to the newly copied Type node.
 5. Attach the Jan-Feb model nugget already in the stream to the Jan model nugget.
 6. Open the Jan model nugget.

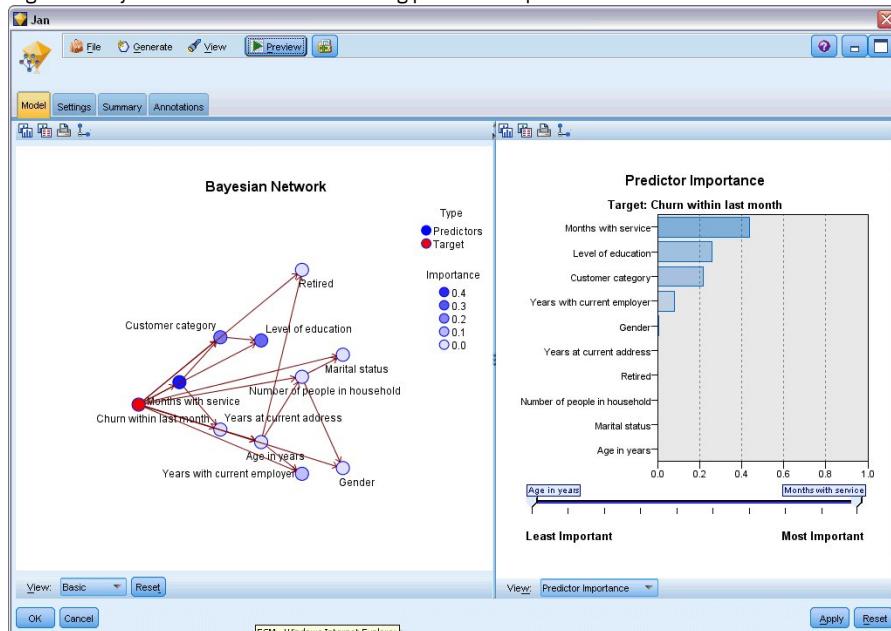
Figure 3. Adding the nuggets to the stream



The Bayesian Network model nugget Model tab is split into two columns. The left column contains a network graph of nodes that displays the relationship between the target and its most important predictors, as well as the relationship between the predictors.

The right column shows either *Predictor Importance*, which indicates the relative importance of each predictor in estimating the model, or *Conditional Probabilities*, which contains the conditional probability value for each node value and each combination of values in its parent nodes.

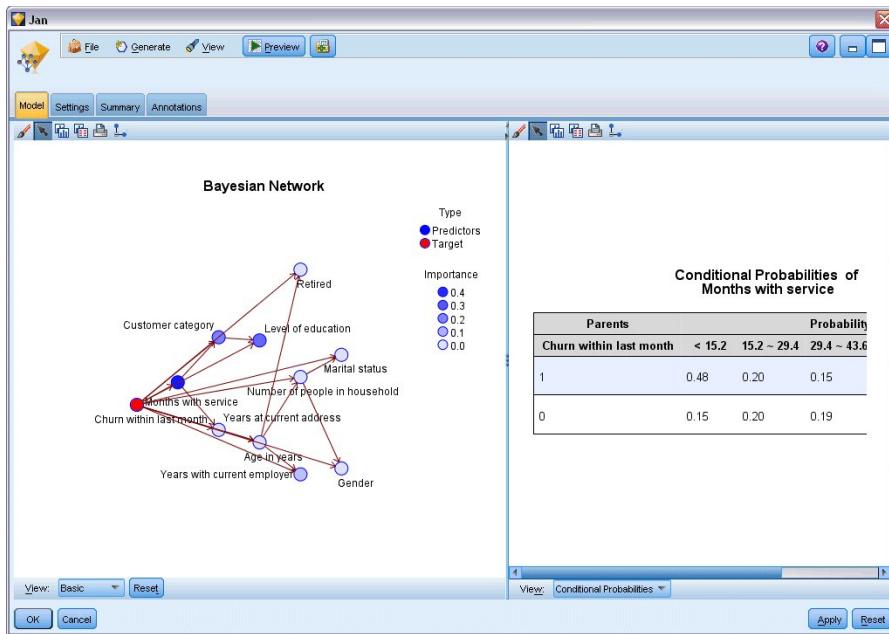
Figure 4. Bayesian Network model showing predictor importance



To display the conditional probabilities for any node, click on the node in the left column. The right column is updated to show the required details.

The conditional probabilities are shown for each bin that the data values have been divided into relative to the node's parent and sibling nodes.

Figure 5. Bayesian Network model showing conditional probabilities



7. To rename the model outputs for clarity, attach a Filter node to the Jan-Feb model nugget.
8. In the right *Field* column, rename \$B-churn as Jan and \$B1-churn as Jan-Feb.

Figure 6. Rename model field names

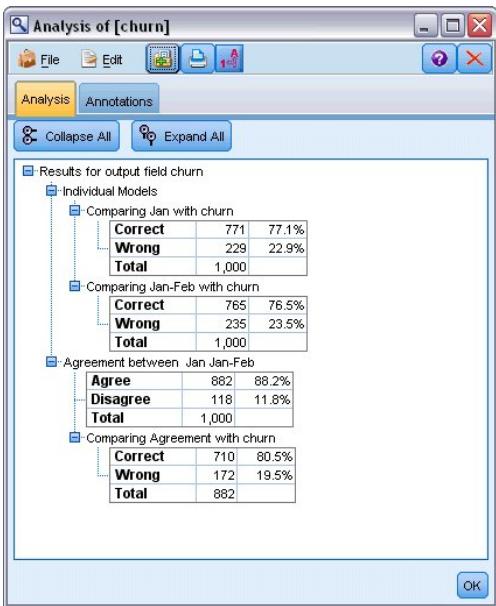


To check how well each model predicts churn, use an Analysis node; this shows the accuracy in terms of percentage for both correct and incorrect predictions.

9. Attach an Analysis node to the Filter node.
10. Open the Analysis node and click Run.

This shows that both models have a similar degree of accuracy when predicting churn.

Figure 7. Analyzing model accuracy



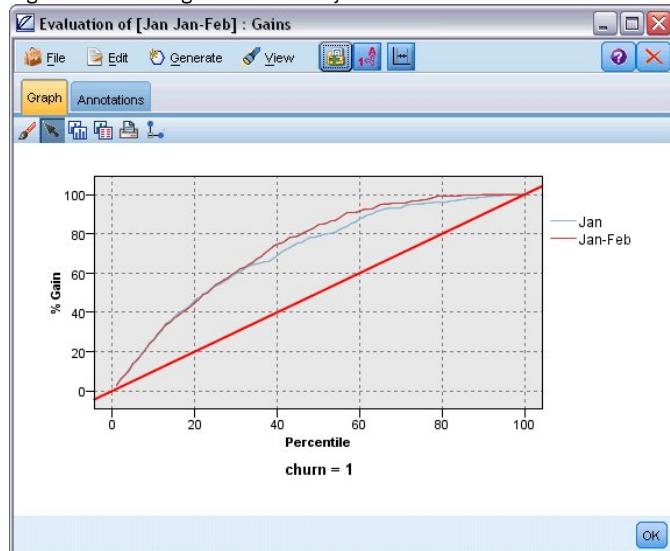
As an alternative to the Analysis node, you can use an Evaluation graph to compare the models' predicted accuracy by building a gains chart.

11. Attach an Evaluation graph node to the Filter node.

and execute the graph node using its default settings.

As with the Analysis node, the graph shows that each model type produces similar results; however, the retrained model using both months' data is slightly better because it has a higher level of confidence in its predictions.

Figure 8. Evaluating model accuracy



Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the *IBM SPSS Modeler Algorithms Guide*, available from the *|Documentation* directory of the installation disk.

Note also that these results are based on the training data only. To assess how well the model generalizes to other data in the real world, you would use a Partition node to hold out a subset of records for purposes of testing and validation.

Retail Sales Promotion (Neural Net/C&RT)

This example deals with data that describes retail product lines and the effects of promotion on sales. (This data is fictitious.) Your goal in this example is to predict the effects of future sales promotions. Similar to the condition monitoring example, the data mining process consists of the exploration, data preparation, training, and test phases.

This example uses the streams named *goodsplot.str* and *goodslearn.str*, which reference the data files named *GOODS1n* and *GOODS2n*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The stream *goodsplot.str* is in the *streams* folder, while the *goodslearn.str* file is in the *streams* directory.

[Next](#)

- [Examining the Data](#)
- [Learning and Testing](#)

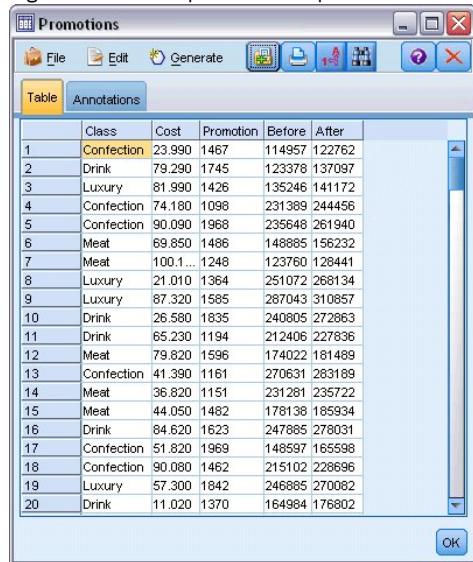
Examining the Data

Each record contains:

- *Class*. Product type.
- *Cost*. Unit price.
- *Promotion*. Index of amount spent on a particular promotion.
- *Before*. Revenue before promotion.
- *After*. Revenue after promotion.

The stream *goodsplot.str* contains a simple stream to display the data in a table. The two revenue fields (*Before* and *After*) are expressed in absolute terms; however, it seems likely that the increase in revenue after the promotion (and presumably as a result of it) would be a more useful figure.

Figure 1. Effects of promotion on product sales



The screenshot shows a software window titled "Promotions". At the top, there's a toolbar with icons for File, Edit, Generate, and others. Below the toolbar, there are two tabs: "Table" (which is selected) and "Annotations". The main area is a table with the following data:

	Class	Cost	Promotion	Before	After
1	Confection	23.990	1467	114957	122762
2	Drink	79.290	1745	123378	137097
3	Luxury	81.990	1426	135246	141172
4	Confection	74.180	1098	231389	244456
5	Confection	90.090	1968	235648	261940
6	Meat	69.850	1486	148885	156232
7	Meat	100.1...	1248	123760	128441
8	Luxury	21.010	1364	251072	268134
9	Luxury	87.320	1585	287043	310857
10	Drink	26.580	1835	240805	272863
11	Drink	65.230	1194	212406	227836
12	Meat	79.820	1596	174022	181489
13	Confection	41.390	1161	270631	283189
14	Meat	36.820	1151	231281	235722
15	Meat	44.050	1482	178138	185934
16	Drink	84.620	1623	247885	278031
17	Confection	51.820	1969	148597	165598
18	Confection	90.080	1462	215102	228696
19	Luxury	57.300	1842	246885	270082
20	Drink	11.020	1370	164984	176802

At the bottom right of the table is an "OK" button.

goodsplot.str also contains a node to derive this value, expressed as a percentage of the revenue before the promotion, in a field called *Increase* and displays a table showing this field.

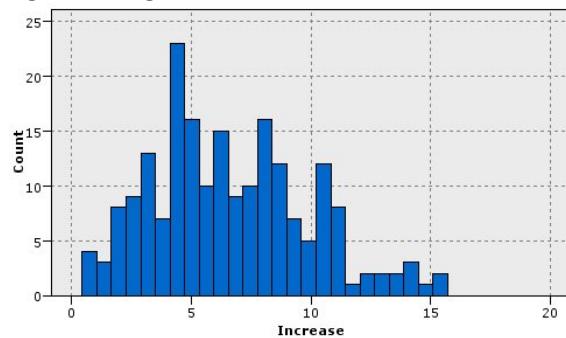
Figure 2. Increase in revenue after promotion

Promotions

	Class	Cost	Promotion	Before	After	Increase
1	Confection	23.990	1467	114957	122762	6.789
2	Drink	79.290	1745	123378	137097	11.119
3	Luxury	81.990	1426	135246	141172	4.382
4	Confection	74.180	1098	231389	244456	5.647
5	Confection	90.090	1968	235648	261940	11.157
6	Meat	69.850	1486	148885	156232	4.935
7	Meat	100.1...	1248	123760	128441	3.782
8	Luxury	21.010	1364	251072	268134	6.796
9	Luxury	87.320	1585	287043	310857	8.296
10	Drink	26.580	1835	240805	272863	13.313
11	Drink	65.230	1194	212408	227836	7.264
12	Meat	79.620	1596	174022	181489	4.291
13	Confection	41.390	1161	270631	283189	4.640
14	Meat	36.620	1151	231281	235722	1.920
15	Meat	44.050	1482	178138	185934	4.376
16	Drink	84.620	1623	247885	278031	12.161
17	Confection	51.820	1969	148597	165598	11.441
18	Confection	90.080	1462	215102	228696	6.320
19	Luxury	57.300	1842	246885	270082	9.396
20	Drink	11.020	1370	164984	176802	7.163

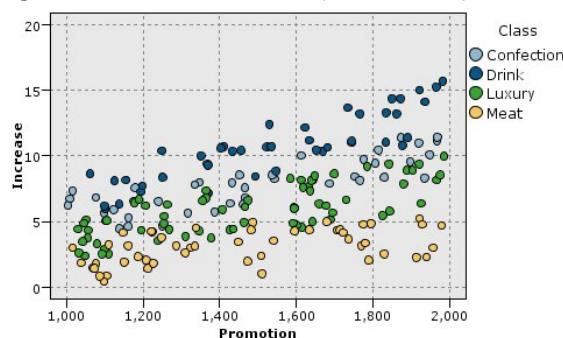
In addition, the stream displays a histogram of the increase and a scatterplot of the increase against the promotion costs expended, overlaid with the category of product involved.

Figure 3. Histogram of increase in revenue



The scatterplot shows that for each class of product, an almost linear relationship exists between the increase in revenue and the cost of promotion. Therefore, it seems likely that a decision tree or neural network could predict, with reasonable accuracy, the increase in revenue from the other available fields.

Figure 4. Revenue increase versus promotional expenditure

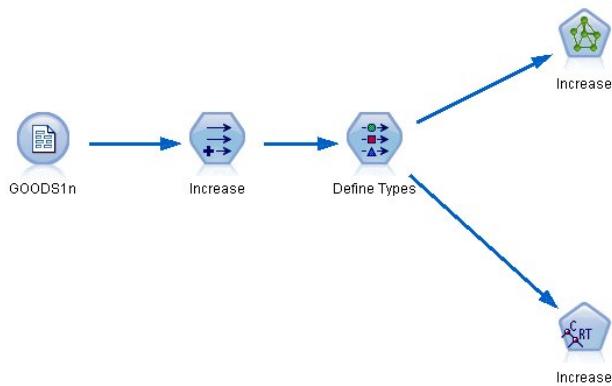


[Next](#)

Learning and Testing

The stream goodslearn.str trains a neural network and a decision tree to make this prediction of revenue increase.

Figure 1. Modeling stream goodslearn.str



Once you have executed the model nodes and generated the actual models, you can test the results of the learning process. You do this by connecting the decision tree and network in series between the Type node and a new Analysis node, changing the input (data) file to GOODS2n, and executing the Analysis node. From the output of this node, in particular from the linear correlation between the predicted increase and the correct answer, you will find that the trained systems predict the increase in revenue with a high degree of success.

Further exploration could focus on the cases where the trained systems make relatively large errors; these could be identified by plotting the predicted increase in revenue against the actual increase. Outliers on this graph could be selected using the interactive graphics within SPSS® Modeler, and from their properties, it might be possible to tune the data description or learning process to improve accuracy.

Condition Monitoring (Neural Net/C5.0)

This example concerns monitoring status information from a machine and the problem of recognizing and predicting fault states. The data is created from a fictitious simulation and consists of a number of concatenated series measured over time. Each record is a snapshot report on the machine in terms of the following:

- *Time*. An integer.
- *Power*. An integer.
- *Temperature*. An integer.
- *Pressure*. 0 if normal, 1 for a momentary pressure warning.
- *Uptime*. Time since last serviced.
- *Status*. Normally 0, changes to error code on error (101, 202, or 303).
- *Outcome*. The error code that appears in this time series, or 0 if no error occurs. (These codes are available only with the benefit of hindsight.)

This example uses the streams named *condplot.str* and *condlearn.str*, which reference the data files named *COND1n* and *COND2n*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *condplot.str* and *condlearn.str* files are in the *streams* directory.

For each time series, there is a series of records from a period of normal operation followed by a period leading to the fault, as shown in the following table:

Time	Power	Temperature	Pressure	Uptime	Status	Outcome
0	1059	259	0	404	0	0
1	1059	259	0	404	0	0
		...				
51	1059	259	0	404	0	0
52	1059	259	0	404	0	0
53	1007	259	0	404	0	303
54	998	259	0	404	0	303
		...				
89	839	259	0	404	0	303
90	834	259	0	404	303	303
0	965	251	0	209	0	0
1	965	251	0	209	0	0
		...				
51	965	251	0	209	0	0
52	965	251	0	209	0	0
53	938	251	0	209	0	101
54	936	251	0	209	0	101

Time	Power	Temperature	Pressure	Uptime	Status	Outcome
			...			
208	644	251	0	209	0	101
209	640	251	0	209	101	101

The following process is common to most data mining projects:

- Examine the data to determine which attributes may be relevant to the prediction or recognition of the states of interest.
- Retain those attributes (if already present), or derive and add them to the data, if necessary.
- Use the resultant data to train rules and neural nets.
- Test the trained systems using independent test data.

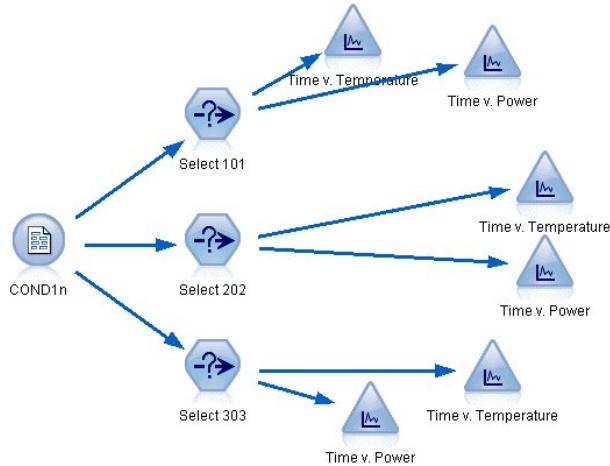
[Next](#)

- [Examining the Data](#)
- [Data Preparation](#)
- [Learning](#)
- [Testing](#)

Examining the Data

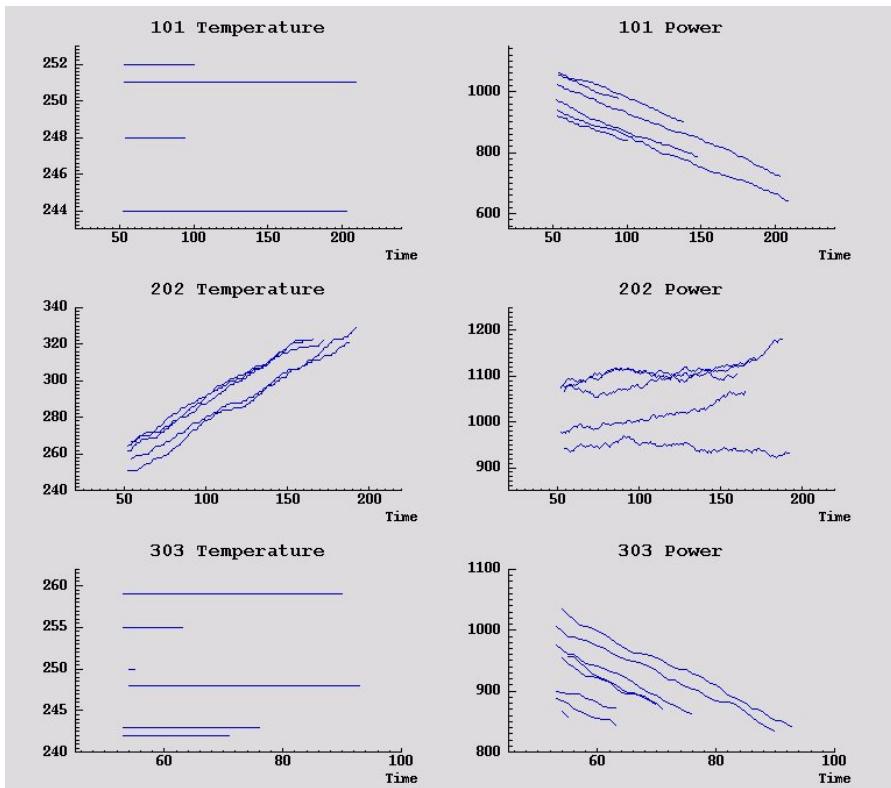
The file *condplot.str* illustrates the first part of the process. It contains a stream that plots a number of graphs. If the time series of temperature or power contains visible patterns, you could differentiate between impending error conditions or possibly predict their occurrence. For both temperature and power, the stream below plots the time series associated with the three different error codes on separate graphs, yielding six graphs. Select nodes separate the data associated with the different error codes.

Figure 1. Condplot stream



The results of this stream are shown in this figure.

Figure 2. Temperature and power over time



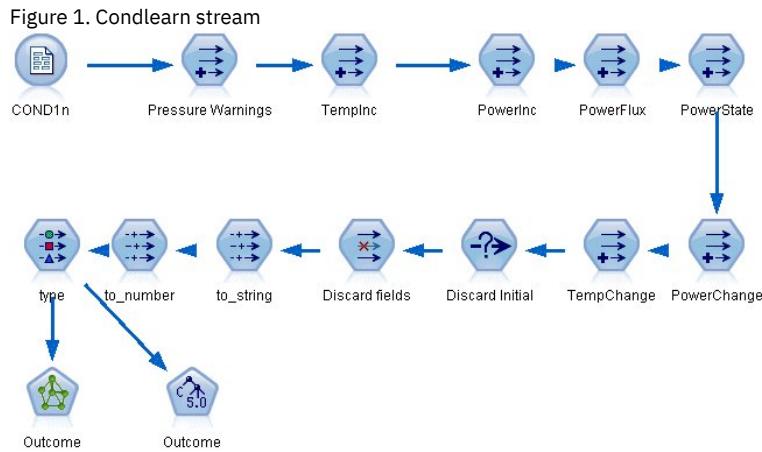
The graphs clearly display patterns distinguishing 202 errors from 101 and 303 errors. The 202 errors show rising temperature and fluctuating power over time; the other errors do not. However, patterns distinguishing 101 from 303 errors are less clear. Both errors show even temperature and a drop in power, but the drop in power seems steeper for 303 errors.

Based on these graphs, it appears that the presence and rate of change for both temperature and power, as well as the presence and degree of fluctuation, are relevant to predicting and distinguishing faults. These attributes should therefore be added to the data before applying the learning systems.

[Next](#)

Data Preparation

Based on the results of exploring the data, the stream *condlearn.str* derives the relevant data and learns to predict faults.



The stream uses a number of Derive nodes to prepare the data for modeling.

- **Variable File node.** Reads data file *COND1n*.
- **Derive Pressure Warnings.** Counts the number of momentary pressure warnings. Reset when time returns to 0.
- **Derive TempInc.** Calculates momentary rate of temperature change using `@DIFF1`.
- **Derive PowerInc.** Calculates momentary rate of power change using `@DIFF1`.

- **Derive PowerFlux**. A flag, true if power varied in opposite directions in the last record and this one; that is, for a power peak or trough.
- **Derive PowerState**. A state that starts as *Stable* and switches to *Fluctuating* when two successive power fluxes are detected. Switches back to *Stable* only when there hasn't been a power flux for five time intervals or when *Time* is reset.
- **PowerChange**. Average of *PowerInc* over the last five time intervals.
- **TempChange**. Average of *TempInc* over the last five time intervals.
- **Discard Initial (select)**. Discards the first record of each time series to avoid large (incorrect) jumps in *Power* and *Temperature* at boundaries.
- **Discard fields**. Cuts records down to *Uptime*, *Status*, *Outcome*, *Pressure Warnings*, *PowerState*, *PowerChange*, and *TempChange*.
- **Type**. Defines the role of *Outcome* as Target (the field to predict). In addition, defines the measurement level of *Outcome* as Nominal, *Pressure Warnings* as Continuous, and *PowerState* as Flag.

[Next](#)

Learning

Running the stream in *condlearn.str* trains the C5.0 rule and neural network (net). The network may take some time to train, but training can be interrupted early to save a net that produces reasonable results. Once the learning is complete, the Models tab at the upper right of the managers window flashes to alert you that two new nuggets were created: one represents the neural net and one represents the rule.

Figure 1. Models manager with model nuggets



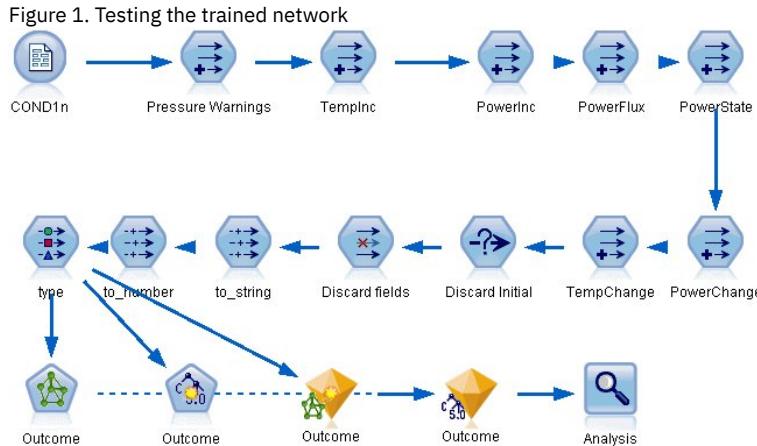
The model nuggets are also added to the existing stream, enabling us to test the system or export the results of the model. In this example, we will test the results of the model.

[Next](#)

Testing

The model nuggets are added to the stream, both of them connected to the Type node.

1. Reposition the nuggets as shown, so that the Type node connects to the neural net nugget, which connects to the C5.0 nugget.
2. Attach an Analysis node to the C5.0 nugget.
3. Edit the original source node to read the file *COND2n* (instead of *COND1n*), as *COND2n* contains unseen test data.



4. Open the Analysis node and click Run.

Doing so yields figures reflecting the accuracy of the trained network and rule.

Classifying Telecommunications Customers (Discriminant Analysis)

Discriminant analysis is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric one.

For example, suppose a telecommunications provider has segmented its customer base by service usage patterns, categorizing the customers into four groups. If demographic data can be used to predict group membership, you can customize offers for individual prospective customers.

This example uses the stream named *telco_custcat_discriminant.str*, which references the data file named *telco.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *telco_custcat_discriminant.str* file is in the *streams* directory.

The example focuses on using demographic data to predict usage patterns. The target field *custcat* has four possible values which correspond to the four customer groups, as follows:

Value	Label
1	Basic Service
2	E-Service
3	Plus Service
4	Total Service

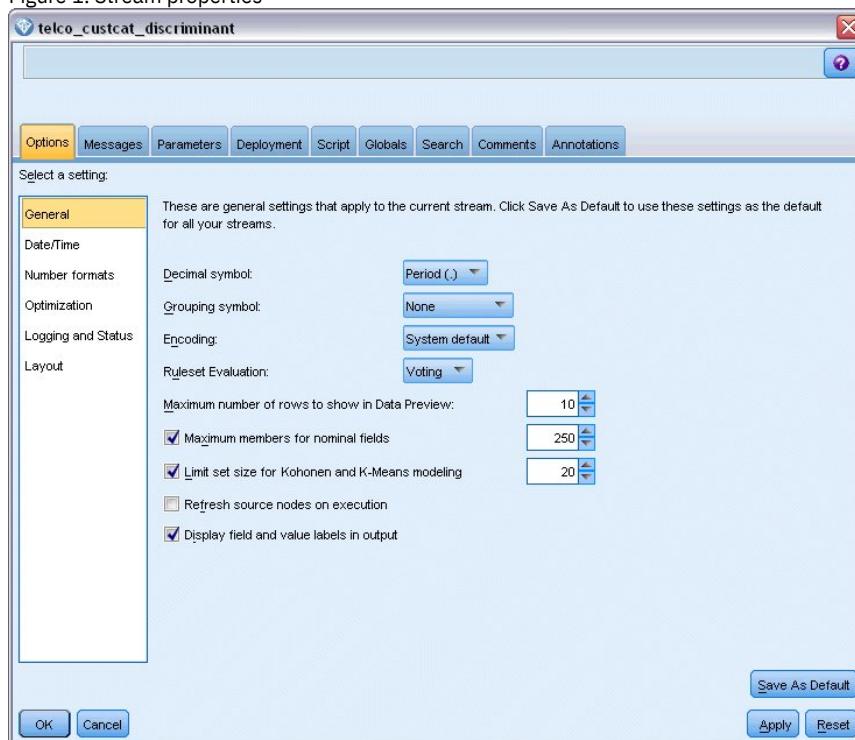
[Next](#)

- [Creating the Stream](#)
- [Examining the Model](#)
- [Summary](#)

Creating the Stream

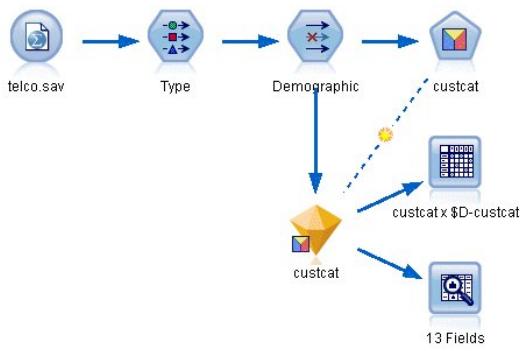
1. First, set the stream properties to show variable and value labels in the output. From the menus, choose:
File > *Stream Properties*... > *Options* > *General*
2. Make sure that *Display field and value labels in output* is selected and click *OK*.

Figure 1. Stream properties



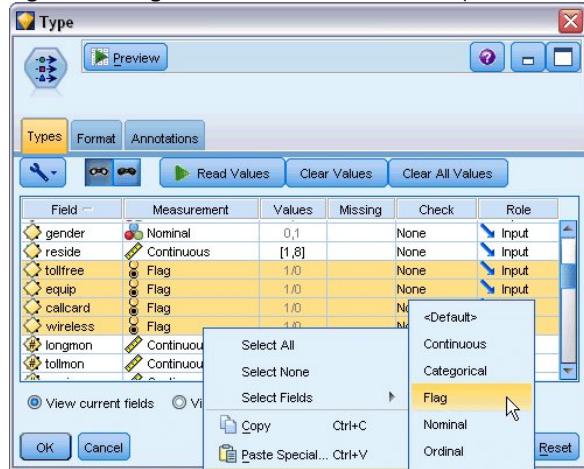
3. Add a Statistics File source node pointing to *telco.sav* in the *Demos* folder.

Figure 2. Sample stream to classify customers using discriminant analysis



- a. Add a Type node and click Read Values, making sure that all measurement levels are set correctly. For example, most fields with values 0 and 1 can be regarded as flags.

Figure 3. Setting the measurement level for multiple fields

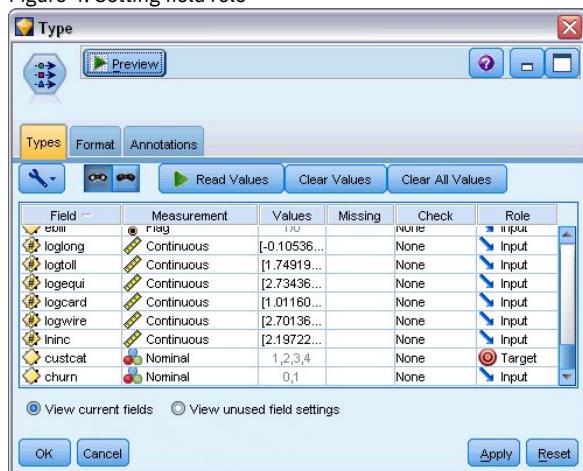


Tip: To change properties for multiple fields with similar values (such as 0/1), click the *Values* column header to sort fields by value, and then hold down the shift key while using the mouse or arrow keys to select all the fields you want to change. You can then right-click on the selection to change the measurement level or other attributes of the selected fields.

Notice that *gender* is more correctly considered as a field with a set of two values, instead of a flag, so leave its Measurement value as Nominal.

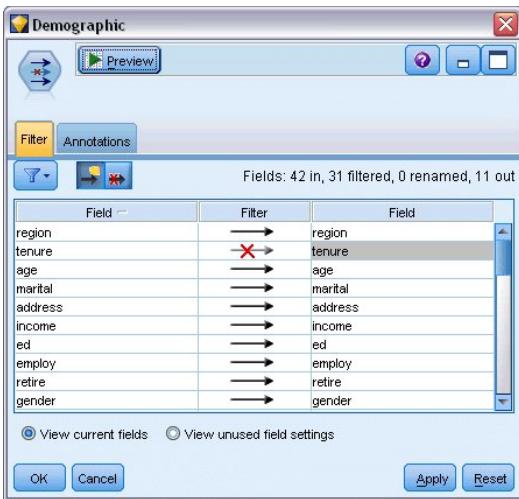
- b. Set the role for the *custcat* field to Target. All other fields should have their role set to Input.

Figure 4. Setting field role



Since this example focuses on demographics, use a Filter node to include only the relevant fields (*region, age, marital, address, income, ed, employ, retire, gender, reside, and custcat*). Other fields can be excluded for the purpose of this analysis.

Figure 5. Filtering on demographic fields



(Alternatively, you could change the role to None for these fields rather than exclude them, or select the fields you want to use in the modeling node.)

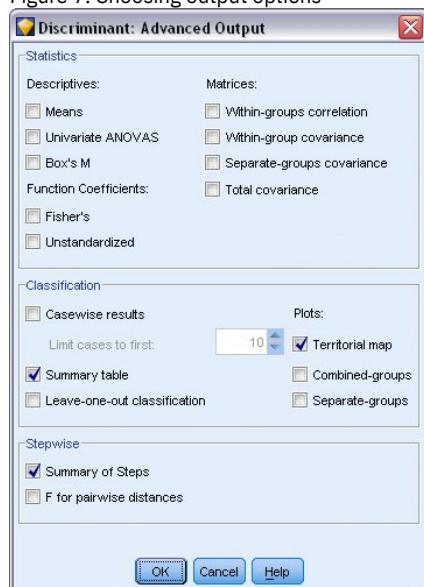
4. In the Discriminant node, click the Model tab and select the Stepwise method.

Figure 6. Choosing model options



5. On the Expert tab, set the mode to Expert and click Output.
6. Select Summary table, Territorial map, and Summary of Steps in the Advanced Output dialog box, then click OK.

Figure 7. Choosing output options

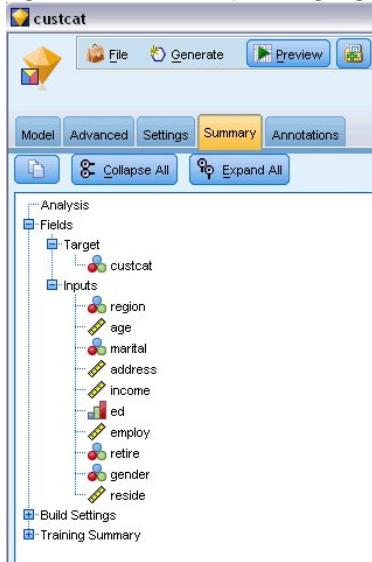


Examining the Model

1. Click Run to create the model, which is added to the stream and to the Models palette in the upper-right corner. To view its details, double-click on the model nugget in the stream.

The Summary tab shows (among other things) the target and the complete list of inputs (predictor fields) submitted for consideration.

Figure 1. Model summary showing target and input fields



For details of the discriminant analysis results:

2. Click the Advanced tab.
3. Click the "Launch in external browser" button (just below the Model tab) to view the results in your Web browser.

- [Analyzing output of using Discriminant analysis to classify telecommunications customers](#)

Analyzing output of using Discriminant analysis to classify telecommunications customers

- [Stepwise Discriminant analysis](#)
- [A note of caution concerning stepwise methods](#)
- [Checking model fit](#)
- [Structure matrix](#)
- [Territorial map](#)
- [Classification results](#)

Stepwise Discriminant analysis

Figure 1. Variables not in the analysis

Variables Not in the Analysis

Step		Tolerance	Min. Tolerance	F to Enter	Wilks' Lambda
0	Age in years	1.000	1.000	7.521	.978
	Marital status	1.000	1.000	3.500	.990
	Years at current address	1.000	1.000	8.433	.975
	Household income in thousands	1.000	1.000	6.689	.980
	Level of education	1.000	1.000	61.454	.844
	Years with current employer	1.000	1.000	16.976	.951
	Retired	1.000	1.000	3.005	.991
	Gender	1.000	1.000	.373	.999
	Number of people in household	1.000	1.000	3.976	.988
	Age in years	.980	.980	6.125	.829
1	Marital status	.999	.999	3.803	.834
	Years at current address	.983	.983	8.487	.823
	Household income in thousands	.989	.989	6.022	.829
	Years with current employer	.953	.953	14.933	.807
	Retired	.992	.992	1.432	.840
	Gender	1.000	1.000	.358	.843
	Number of people in household	1.000	1.000	3.967	.834
	Age in years	.563	.548	.352	.807
	Marital status	.999	.952	3.903	.798

When you have a lot of predictors, the stepwise method can be useful by automatically selecting the "best" variables to use in the model. The stepwise method starts with a model that doesn't include any of the predictors. At each step, the predictor with the largest *F to Enter* value that exceeds the entry criteria (by default, 3.84) is added to the model.

The variables left out of the analysis at the last step all have *F to Enter* values smaller than 3.84, so no more are added.

Figure 2. Variables in the analysis

Variables in the Analysis

Step		Tolerance	F to Remove	Wilks' Lambda
1	Level of education	1.000	61.454	
	Level of education	.953	59.108	.951
	Years with current employer	.953	14.933	.844
	Level of education	.951	60.046	.940
	Years with current employer	.934	15.824	.834
	Number of people in household	.979	4.841	.807

This table displays statistics for the variables that are in the analysis at each step. *Tolerance* is the proportion of a variable's variance not accounted for by other independent variables in the equation. A variable with very low tolerance contributes little information to a model and can cause computational problems.

F to Remove values are useful for describing what happens if a variable is removed from the current model (given that the other variables remain). *F to Remove* for the entering variable is the same as *F to Enter* at the previous step (shown in the *Variables Not in the Analysis* table).

[Next](#)

A note of caution concerning stepwise methods

Stepwise methods are convenient, but have their limitations. Be aware that because stepwise methods select models based solely upon statistical merit, it may choose predictors that have no *practical significance*. If you have some experience with the data and have expectations about which predictors are important, you should use that knowledge and eschew stepwise methods. If, however, you have many predictors and no idea where to start, running a stepwise analysis and adjusting the selected model is better than no model at all.

[Next](#)

Checking model fit

Figure 1. Eigenvalues

Eigenvalues				
Function	Eigenvalue	% of Variance	Cumulative %	Canonical Correlation
1	.198 ^a	80.2	80.2	.407
2	.048 ^a	19.4	99.6	.214
3	.001 ^a	.4	100.0	.031

a.First 3 canonical discriminant functions were used in the analysis.

Nearly all of the variance explained by the model is due to the first two discriminant functions. Three functions are fit automatically, but due to its minuscule eigenvalue, you can fairly safely ignore the third.

Figure 2. Wilks' lambda

Wilks' Lambda				
Test of Function(s)	Wilks' Lambda	Chi-square	df	Sig.
1 through 3	.796	227.345	9	<.001
2 through 3	.953	47.486	4	<.001
3	.999	.929	1	.335

Wilks' lambda agrees that only the first two functions are useful. For each set of functions, this tests the hypothesis that the means of the functions listed are equal across groups. The test of function 3 has a significance value greater than 0.10, so this function contributes little to the model.

[Next](#)

Structure matrix

Figure 1. Structure matrix

Structure Matrix

	Function		
	1	2	3
Level of education	.966*	-.090	-.244
Years with current employer	-.182	.964*	-.193
Age in years^b	-.162	.598*	-.285
Household income in thousands^b	.109	.514*	-.190
Years at current address^b	-.151	.394*	-.214
Retired^b	-.108	.230*	-.137
Gender^b	.008	.054*	.009
Number of people in household	.232	.097	.968*
Marital status^b	.132	.134	.600*

Pooled within-groups correlations between discriminating variables and standardized canonical discriminant functions
Variables ordered by absolute size of correlation within function.

*.Largest absolute correlation between each variable and any discriminant function

b.This variable not used in the analysis.

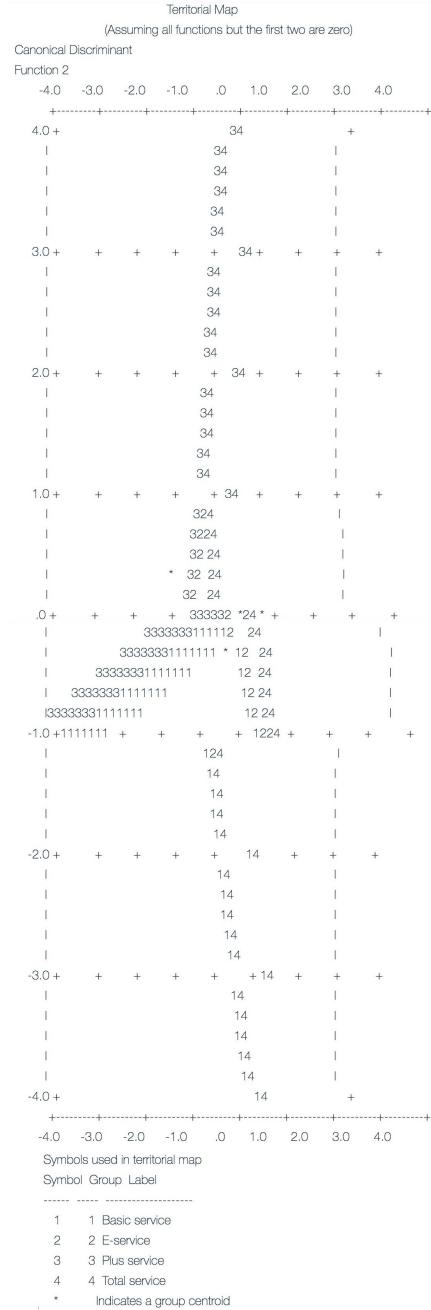
When there is more than one discriminant function, an asterisk(*) marks each variable's largest absolute correlation with one of the canonical functions. Within each function, these marked variables are then ordered by the size of the correlation.

- *Level of education* is most strongly correlated with the first function, and it is the only variable most strongly correlated with this function.
- *Years with current employer*, *Age in years*, *Household income in thousands*, *Years at current address*, *Retired*, and *Gender* are most strongly correlated with the second function, although *Gender* and *Retired* are more weakly correlated than the others. The other variables mark this function as a "stability" function.
- *Number of people in household* and *Marital status* are most strongly correlated with the third discriminant function, but this is a useless function, so these are nearly useless predictors.

[Next](#)

Territorial map

Figure 1. Territorial map



The territorial map helps you to study the relationships between the groups and the discriminant functions. Combined with the structure matrix results, it gives a graphical interpretation of the relationship between predictors and groups. The first function, shown on the horizontal axis, separates group 4 (*Total service customers*) from the others. Since *Level of education* is strongly positively correlated with the first function, this suggests that your *Total service customers* are, in general, the most highly educated. The second function separates groups 1 and 3 (*Basic service* and *Plus service customers*). *Plus service* customers tend to have been working longer and are older than *Basic service* customers. *E-service* customers are not separated well from the others, although the map suggests that they tend to be well educated with a moderate amount of work experience.

In general, the closeness of the group centroids, marked with asterisks (*), to the territorial lines suggests that the separation between all groups is not very strong.

Only the first two discriminant functions are plotted, but since the third function was found to be rather insignificant, the territorial map offers a comprehensive view of the discriminant model.

[Next](#)

Classification results

Figure 1. Classification results

		Classification Results ^a					Total	
		Customer category	Basic service	E-service	Plus service	Total service		
Original	Count	Basic service	125	11	61	69	266	
		E-service	49	15	58	95	217	
		Plus service	102	14	112	53	281	
	%	Total service	40	16	37	143	236	
		Basic service	47.0	4.1	22.9	25.9	100.0	
		E-service	22.6	6.9	26.7	43.8	100.0	
		Plus service	36.3	5.0	39.9	18.9	100.0	
		Total service	16.9	6.8	15.7	60.6	100.0	

^a39.5% of original grouped cases correctly classified.

From Wilks' lambda, you know that your model is doing better than guessing, but you need to turn to the classification results to determine how much better. Given the observed data, the "null" model (that is, one without predictors) would classify all customers into the modal group, *Plus service*. Thus, the null model would be correct 281/1000 = 28.1% of the time. Your model gets 11.4% more or 39.5% of the customers. In particular, your model excels at identifying *Total service* customers. However, it does an exceptionally poor job of classifying *E-service* customers. You may need to find another predictor in order to separate these customers.

[Next](#)

Summary

You have created a discriminant model that classifies customers into one of four predefined "service usage" groups, based on demographic information from each customer. Using the structure matrix and territorial map, you identified which variables are most useful for segmenting your customer base. Lastly, the classification results show that the model does poorly at classifying *E-service* customers. More research is required to determine another predictor variable that better classifies these customers, but depending on what you are looking to predict, the model may be perfectly adequate for your needs. For example, if you are not concerned with identifying *E-service* customers the model may be accurate enough for you. This may be the case where the *E-service* is a loss-leader which brings in little profit. If, for example, your highest return on investment comes from *Plus service* or *Total service* customers, the model may give you the information you need.

Also note that these results are based on the training data only. To assess how well the model generalizes to other data, you can use a Partition node to hold out a subset of records for purposes of testing and validation.

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the IBM SPSS Modeler Algorithms Guide. This is available from the |Documentation directory of the installation disk.

Analyzing interval-censored survival data (Generalized Linear Models)

When analyzing survival data with interval censoring—that is, when the exact time of the event of interest is not known but is known only to have occurred within a given interval—then applying the Cox model to the hazards of events in intervals results in a complementary log-log regression model.

Partial information from a study designed to compare the efficacy of two therapies for preventing the recurrence of ulcers is collected in *ulcer_recurrence.sav*. This dataset has been presented and analyzed elsewhere¹. Using generalized linear models, you can replicate the results for the complementary log-log regression models.

This example uses the stream named *ulcer_genlin.str*, which references the data file *ulcer_recurrence.sav*. The data file is in the *Demos* folder and the stream file is in the *streams* subfolder.

[Next](#)

- [Creating the Stream](#)
- [Tests of model effects](#)
- [Fitting the treatment-only model](#)
- [Parameter estimates](#)
- [Predicted recurrence and survival probabilities](#)
- [Modeling the recurrence probability by period](#)
- [Tests of model effects](#)
- [Fitting the reduced model](#)
- [Parameter estimates](#)

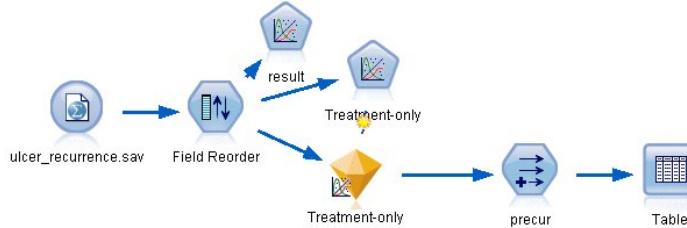
- [Predicted recurrence and survival probabilities](#)
- [Summary](#)
- [Related procedures](#)
- [Recommended readings](#)

¹ Collett, D. 2003. *Modelling survival data in medical research*, 2 ed. Boca Raton: Chapman & Hall/CRC.

Creating the Stream

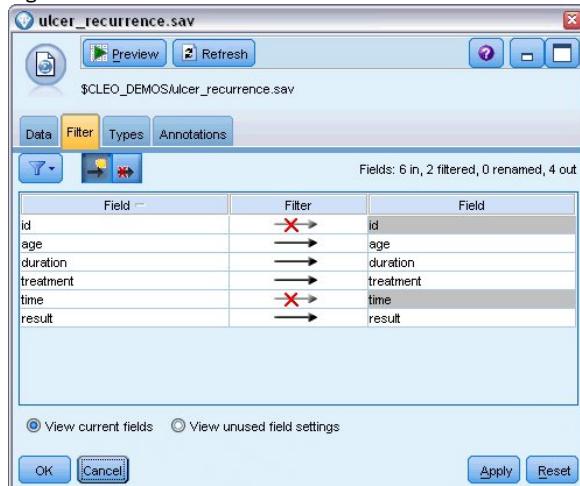
1. Add a Statistics File source node pointing to *ulcer_recurrence.sav* in the *Demos* folder.

Figure 1. Sample stream to predict ulcer recurrence



2. On the Filter tab of the source node, filter out *id* and *time*.

Figure 2. Filter unwanted fields



3. On the Types tab of the source node, set the role for the *result* field to Target and set its measurement level to Flag. A result of 1 indicates that the ulcer has recurred. All other fields should have their role set to Input.
4. Click Read Values to instantiate the data.

Figure 3. Setting field role



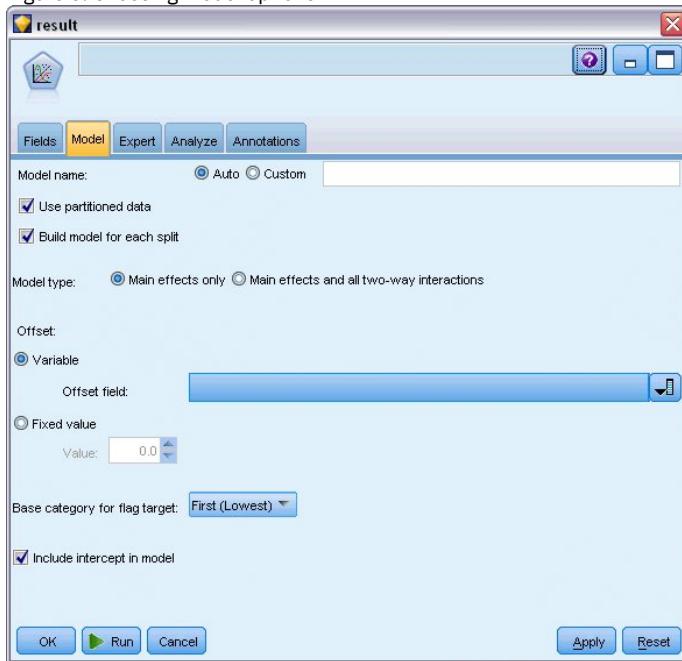
5. Add a Field Reorder node and specify *duration*, *treatment*, and *age* as the order of inputs. This determines the order in which fields are entered in the model and will help you try to replicate Collett's results.

Figure 4. Reordering fields so they are entered into the model as desired



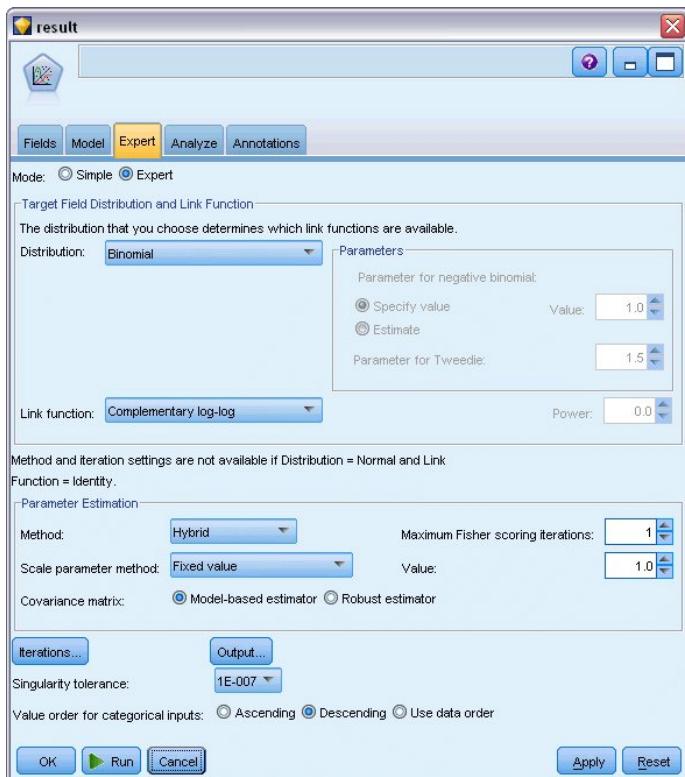
6. Attach a GenLin node to the source node; on the GenLin node, click the Model tab.
 7. Select First (Lowest) as the reference category for the target. This indicates that the second category is the event of interest, and its effect on the model is in the interpretation of parameter estimates. A continuous predictor with a positive coefficient indicates increased probability of recurrence with increasing values of the predictor; categories of a nominal predictor with larger coefficients indicate increased probability of recurrence with respect to other categories of the set.

Figure 5. Choosing model options



8. Click the Expert tab and select Expert to activate the expert modeling options.
 9. Select Binomial as the distribution and Complementary log-log as the link function.
 10. Select Fixed value as the method for estimating the scale parameter and leave the default value of 1.0.
 11. Select Descending as the category order for factors. This indicates that the first category of each factor will be its reference category; the effect of this selection on the model is in the interpretation of parameter estimates.

Figure 6. Choosing expert options



12. Run the stream to create the model nugget, which is added to the stream canvas, and also to the Models palette in the upper right corner. To view the model details, right-click the nugget and choose Edit or Browse.

[Next](#)

Tests of model effects

Figure 1. Tests of model effects for main-effects model

Tests of Model Effects

Source	Type III		
	Wald Chi-Square	df	Sig.
(Intercept)	.536	1	.464
Age in years	.358	1	.550
Duration of disease	.003	1	.958
Treatment group	.382	1	.537

Dependent Variable: Result

Model: (Intercept), Age in years, Duration of disease, Treatment group

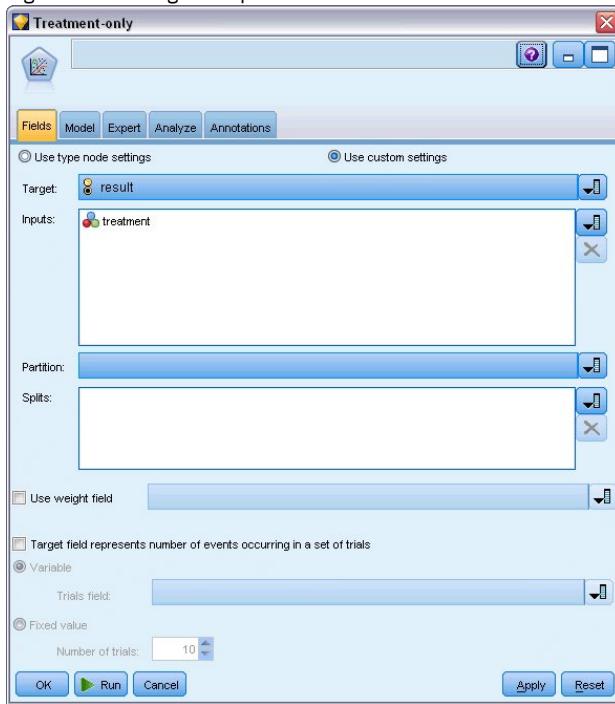
None of the model effects is statistically significant; however, any observable differences in the treatment effects are of clinical interest, so we will fit a reduced model with just the treatment as a model term.

[Next](#)

Fitting the treatment-only model

1. On the Fields tab of the GenLin node, click Use custom settings.
2. Select *result* as the target.
3. Select *treatment* as the sole input.

Figure 1. Choosing field options



4. Run the stream and open the resulting model nugget.

On the model nugget, select the Advanced tab and scroll to the bottom.

[Next](#)

Parameter estimates

Figure 1. Parameter estimates for treatment-only model

Parameter Estimates

Parameter	B	Std. Error	95% Wald Confidence Interval		Hypothesis Test		
			Lower	Upper	Wald Chi-Square	df	Sig.
(Intercept)	-1.442	.5012	-2.425	-.460	8.282	1	.004
[Treatment group=1]	.378	.6288	-.855	1.610	.361	1	.548
[Treatment group=0]	0 ^a
(Scale)	1 ^b						

Dependent Variable: Result

Model: (Intercept), Treatment group

a. Set to zero because this parameter is redundant.

b. Fixed at the displayed value.

The treatment effect (the difference of the linear predictor between the two treatment levels; that is, the coefficient for *treatment=1*) is still not statistically significant, but only suggestive that treatment A [*treatment=0*] may be better than B [*treatment=1*] because the parameter estimate for treatment B is larger than that for A, and is thus associated with an increased probability of recurrence in the first 12 months. The linear predictor, (intercept + treatment effect) is an estimate of $\log(-\log(1-P(\text{recur}_{12,t})))$, where $P(\text{recur}_{12,t})$ is the probability of recurrence at 12 months for treatment t(A or B). These predicted probabilities are generated for each observation in the dataset.

[Next](#)

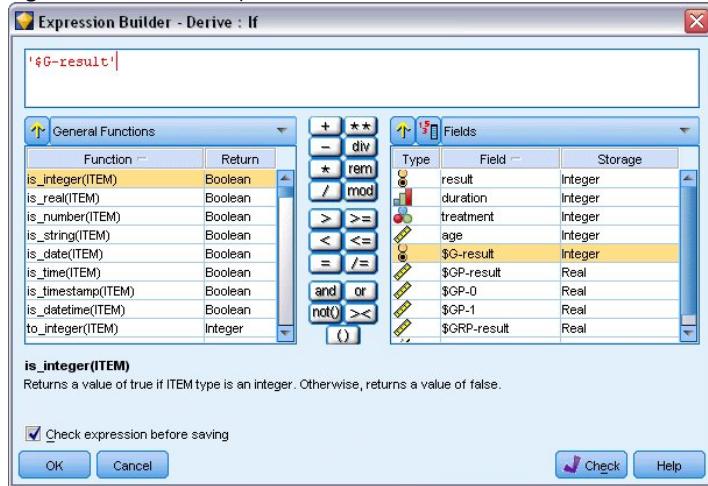
Predicted recurrence and survival probabilities

Figure 1. Derive node settings options



1. For each patient, the model scores the predicted result and the probability of that predicted result. In order to see the predicted recurrence probabilities, copy the generated model to the palette and attach a Derive node.
2. In the Settings tab, type *precur* as the derive field.
3. Choose to derive it as Conditional.
4. Click the calculator button to open the Expression Builder for the If condition.

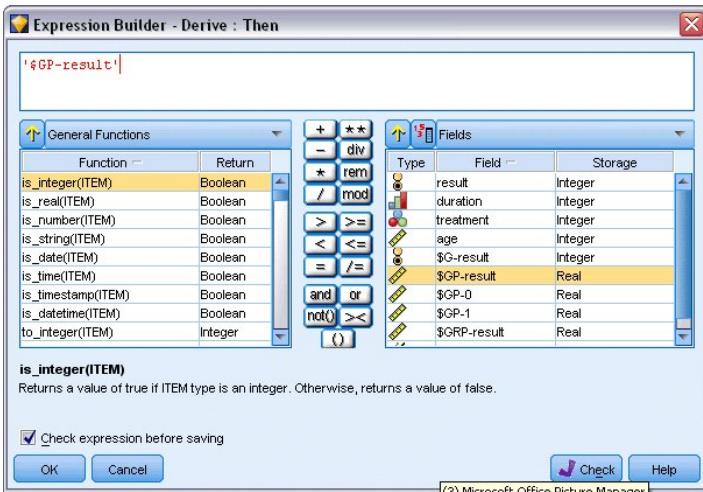
Figure 2. Derive node: Expression Builder for If condition



5. Insert the *\$G-result* field into the expression.
6. Click OK.

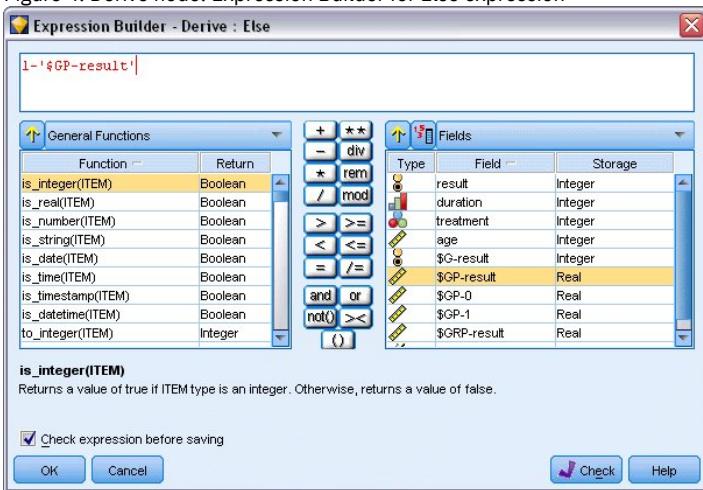
The derive field *precur* will take the value of the Then expression when *\$G-result* equals 1 and the value of the Else expression when it is 0.

Figure 3. Derive node: Expression Builder for Then expression



7. Click the calculator button to open the Expression Builder for the Then expression.
8. Insert the \$GP-result field into the expression.
9. Click OK.

Figure 4. Derive node: Expression Builder for Else expression



10. Click the calculator button to open the Expression Builder for the Else expression.
11. Type 1 – in the expression and then insert the \$GP-result field into the expression.
12. Click OK.

Figure 5. Derive node settings options



13. Attach a table node to the Derive node and execute it.

Figure 6. Predicted probabilities

	result	duration	treatment	age	\$G-result	\$GP-result	\$GP-0	\$GP-1
1	1	2	1	48	0	0.708	0.708	0.292
2	0	1	1	73	0	0.708	0.708	0.292
3	0	1	1	54	0	0.708	0.708	0.292
4	0	2	1	58	0	0.708	0.708	0.292
5	0	1	0	56	0	0.789	0.789	0.211
6	0	2	0	49	0	0.789	0.789	0.211
7	0	1	1	71	0	0.708	0.708	0.292
8	0	1	0	41	0	0.789	0.789	0.211
9	0	1	1	23	0	0.708	0.708	0.292
10	1	1	1	37	0	0.708	0.708	0.292
11	0	1	1	38	0	0.708	0.708	0.292
12	0	2	1	76	0	0.708	0.708	0.292
13	0	2	0	38	0	0.789	0.789	0.211
14	1	1	0	27	0	0.789	0.789	0.211
15	1	1	1	47	0	0.708	0.708	0.292
16	0	1	0	54	0	0.789	0.789	0.211
17	1	1	1	38	0	0.708	0.708	0.292
18	1	2	1	27	0	0.708	0.708	0.292
19	0	2	0	58	0	0.789	0.789	0.211
20	0	1	1	75	0	0.708	0.708	0.292

There is an estimated 0.211 probability that patients assigned to treatment A will experience a recurrence in the first 12 months; 0.292 for treatment B. Note that $1 - P(\text{recur}_{12}, t)$ is the survivor probability at 12 months, which may be of more interest to survival analysts.

[Next](#)

Modeling the recurrence probability by period

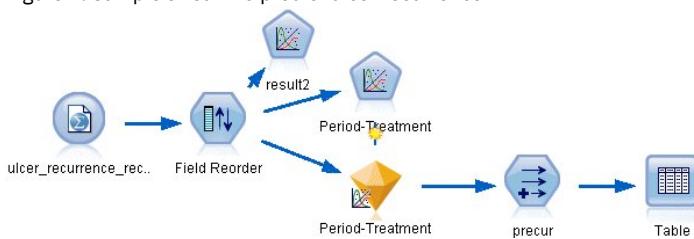
A problem with the model as it stands is that it ignores the information gathered at the first examination; that is, that many patients did not experience a recurrence in the first six months. A "better" model would model a binary response that records whether or not the event occurred during each interval. Fitting this model requires a reconstruction of the original dataset, which can be found in *ulcer_recurrence_recoded.sav*. This file contains two additional variables:

- *Period*, which records whether the case corresponds to the first examination period or the second.
- *Result by period*, which records whether there was a recurrence for the given patient during the given period.

Each original case (patient) contributes one case per interval in which it remains in the risk set. Thus, for example, patient 1 contributes two cases; one for the first examination period in which no recurrence occurred, and one for the second examination period, in which a recurrence was recorded. Patient 10, on the other hand, contributes a single case because a recurrence was recorded in the first period. Patients 16, 28, and 34 dropped out of the study after six months, and thus contribute only a single case to the new dataset.

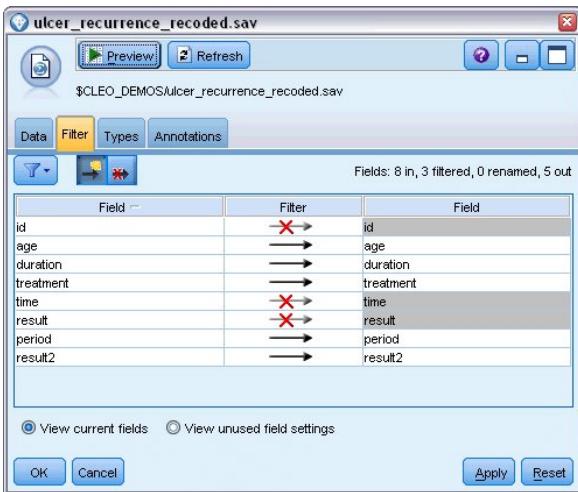
1. Add a Statistics File source node pointing to *ulcer_recurrence_recoded.sav* in the *Demos* folder.

Figure 1. Sample stream to predict ulcer recurrence



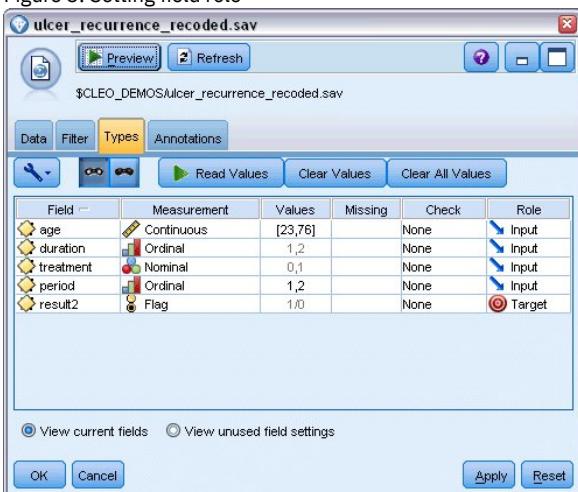
2. On the Filter tab of the source node, filter out *id*, *time*, and *result*.

Figure 2. Filter unwanted fields



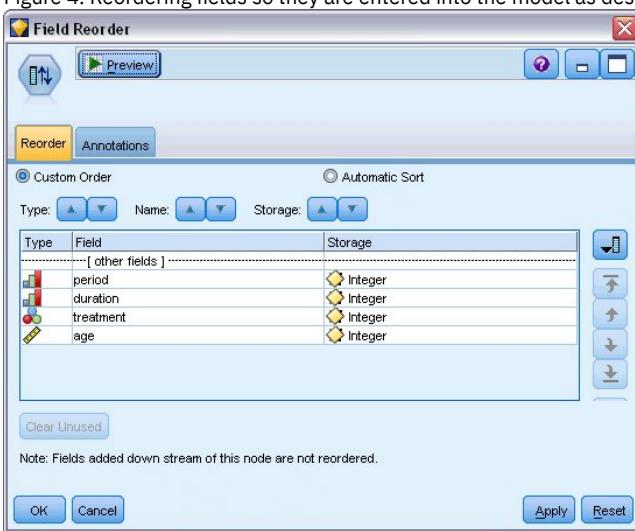
3. On the Types tab of the source node, set the role for the *result2* field to Target and set its measurement level to Flag. All other fields should have their role set to Input.

Figure 3. Setting field role



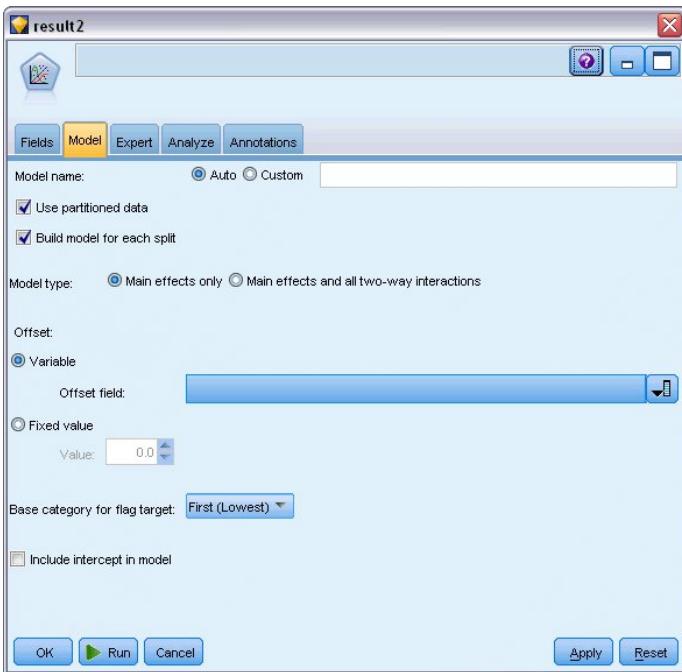
4. Add a Field Reorder node and specify *period*, *duration*, *treatment*, and *age* as the order of inputs. Making *period* the first input (and not including the intercept term in the model) will allow you to fit a full set of dummy variables to capture the period effects.

Figure 4. Reordering fields so they are entered into the model as desired



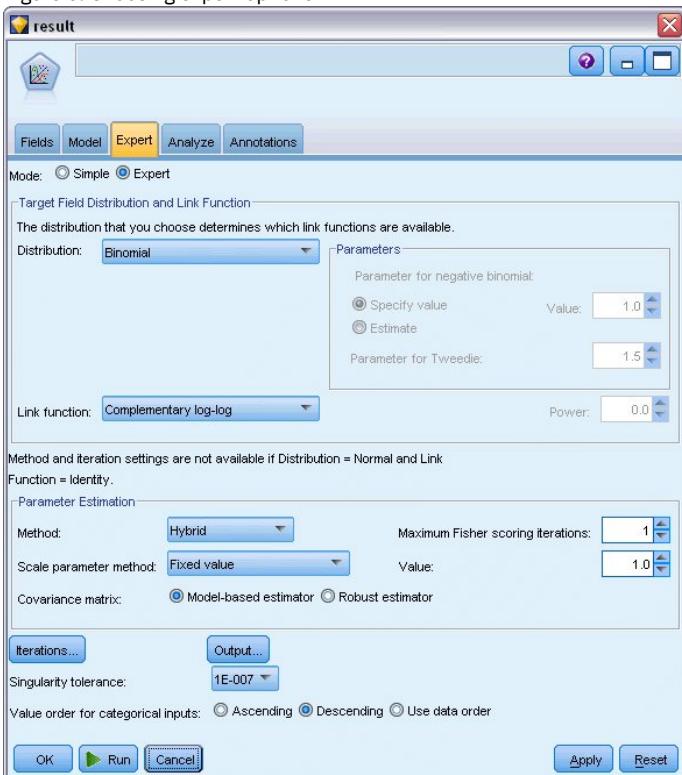
5. On the GenLin node, click the Model tab.

Figure 5. Choosing model options



6. Select First (Lowest) as the reference category for the target. This indicates that the second category is the event of interest, and its effect on the model is in the interpretation of parameter estimates.
7. Deselect Include intercept in model.
8. Click the Expert tab and select Expert to activate the expert modeling options.

Figure 6. Choosing expert options



9. Select Binomial as the distribution and Complementary log-log as the link function.
10. Select Fixed value as the method for estimating the scale parameter and leave the default value of 1.0.
11. Select Descending as the category order for factors. This indicates that the first category of each factor will be its reference category; the effect of this selection on the model is in the interpretation of parameter estimates.
12. Run the stream to create the model nugget, which is added to the stream canvas, and also to the Models palette in the upper right corner. To view the model details, right-click the nugget and choose Edit or Browse.

[Next](#)

Tests of model effects

Figure 1. Tests of model effects for main-effects model

Tests of Model Effects

Source	Type III		
	Wald Chi-Square	df	Sig.
Period	.464	1	.496
Age in years	.314	1	.575
Duration of disease	.000	1	.988
Treatment group	.117	1	.732

Dependent Variable: Result by period

Model: Period, Age in years, Duration of disease, Treatment group

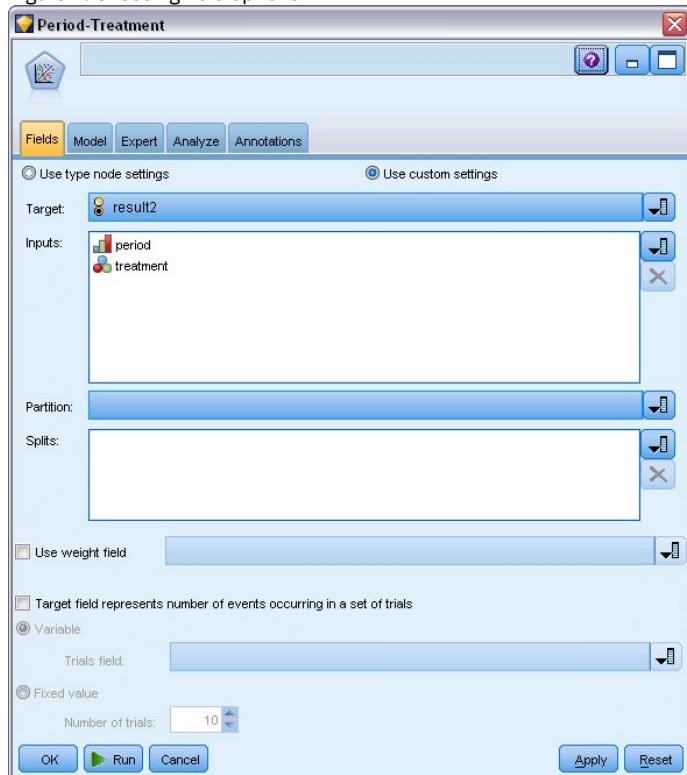
None of the model effects is statistically significant; however, any observable differences in the period and treatment effects are of clinical interest, so we will fit a reduced model with just those model terms.

[Next](#)

Fitting the reduced model

1. On the Fields tab of the GenLin node, click Use custom settings.
2. Select *result2* as the target.
3. Select *period* and *treatment* as the inputs.

Figure 1. Choosing field options



4. Execute the node and browse the generated model, and then copy the generated model to the palette, attach a table node, and execute it.

[Next](#)

Parameter estimates

Figure 1. Parameter estimates for treatment-only model

Parameter Estimates

Parameter	B	Std. Error	95% Wald Confidence Interval		Hypothesis Test		
			Lower	Upper	Wald Chi-Square	df	Sig.
[Period=2]	-1.794	.5792	-2.929	-.659	9.597	1	.002
[Period=1]	-2.206	.5912	-3.365	-1.047	13.926	1	.000
[Treatment group=1]	.195	.6279	-1.035	1.426	.097	1	.756
[Treatment group=0]	0 ^a
(Scale)	1 ^b						

Dependent Variable: Result by period

Model: Period, Treatment group

a. Set to zero because this parameter is redundant.

b. Fixed at the displayed value.

The treatment effect is still not statistically significant but only suggestive that treatment A may be better than B because the parameter estimate for treatment B is associated with an increased probability of recurrence in the first 12 months. The period values are statistically significantly different from 0, but this is because of the fact that an intercept term is not fit. The period effect (the difference between the values of the linear predictor for [period=1] and [period=2]) is not statistically significant, as can be seen in the tests of model effects. The linear predictor (period effect + treatment effect) is an estimate of $\log(-\log(1-P(\text{recur}_p, t)))$, where $P(\text{recur}_p, t)$ is the probability of recurrence at the period p (=1 or 2, representing six months or 12 months) given treatment t (=A or B). These predicted probabilities are generated for each observation in the dataset.

[Next](#)

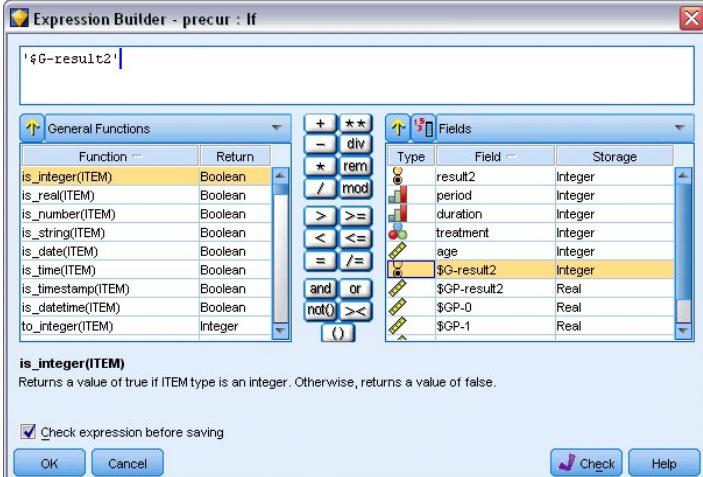
Predicted recurrence and survival probabilities

Figure 1. Derive node settings options



- For each patient, the model scores the predicted result and the probability of that predicted result. In order to see the predicted recurrence probabilities, copy the generated model to the palette and attach a Derive node.
- In the Settings tab, type precur as the derive field.
- Choose to derive it as Conditional.
- Click the calculator button to open the Expression Builder for the If condition.

Figure 2. Derive node: Expression Builder for If condition

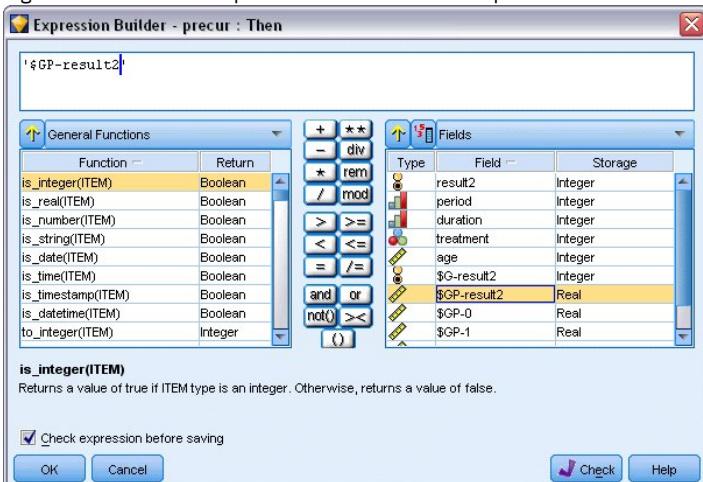


5. Insert the \$G-result2 field into the expression.

6. Click OK.

The derive field *precur* will take the value of the Then expression when \$G-result2 equals 1 and the value of the Else expression when it is 0.

Figure 3. Derive node: Expression Builder for Then expression

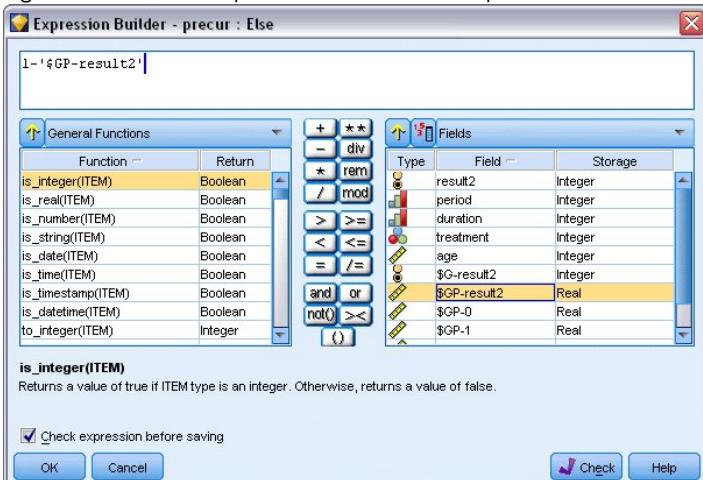


7. Click the calculator button to open the Expression Builder for the Then expression.

8. Insert the \$GP-result2 field into the expression.

9. Click OK.

Figure 4. Derive node: Expression Builder for Else expression



10. Click the calculator button to open the Expression Builder for the Else expression.

11. Type 1 – in the expression and then insert the \$GP-result2 field into the expression.

12. Click OK.

Figure 5. Derive node settings options



13. Attach a table node to the Derive node and execute it.

Figure 6. Predicted probabilities

	result2	period	duration	treatment	age	\$G-result2	\$GP-result2	\$GP-0	\$GP-1
1	0	1	2	1	48	0	0.875	0.875	0.125
2	1	2	2	1	48	0	0.817	0.817	0.183
3	0	1	1	1	73	0	0.875	0.875	0.125
4	0	2	1	1	73	0	0.817	0.817	0.183
5	0	1	1	1	54	0	0.875	0.875	0.125
6	0	2	1	1	54	0	0.817	0.817	0.183
7	0	1	2	1	58	0	0.875	0.875	0.125
8	0	2	2	1	58	0	0.817	0.817	0.183
9	0	1	1	0	56	0	0.896	0.896	0.104
10	0	2	1	0	56	0	0.847	0.847	0.153
11	0	1	2	0	49	0	0.896	0.896	0.104
12	0	2	2	0	49	0	0.847	0.847	0.153
13	0	1	1	1	71	0	0.875	0.875	0.125
14	0	2	1	1	71	0	0.817	0.817	0.183
15	0	1	1	0	41	0	0.896	0.896	0.104
16	0	2	1	0	41	0	0.847	0.847	0.153
17	0	1	1	1	23	0	0.875	0.875	0.125
18	0	2	1	1	23	0	0.817	0.817	0.183
19	1	1	1	1	37	0	0.875	0.875	0.125
20	0	1	1	1	38	0	0.875	0.875	0.125

Table 1. Estimated recurrence probabilities

Treatment	6 months	12 months
A	0.104	0.153
B	0.125	0.183

From the estimated recurrence probabilities, the survival probability through 12 months can be estimated as $1 - (P(\text{recur}_1, t) + P(\text{recur}_2, t)) \times (1 - P(\text{recur}_1, t))$; thus, for each treatment:

$$A: 1 - (0.104 + 0.153 \times 0.896) = 0.759$$

$$B: 1 - (0.125 + 0.183 \times 0.875) = 0.715$$

which again shows nonstatistically significant support for A as the better treatment.

[Next](#)

Summary

Using Generalized Linear Models, you have fit a series of complementary log-log regression models for interval-censored survival data. While there is some support for choosing treatment A, achieving a statistically significant result may require a larger study. However, there are some further avenues to explore with the existing data.

- It may be worthwhile to refit the model with interaction effects, particularly between *Period* and *Treatment group*.

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the *IBM SPSS Modeler Algorithms Guide*.

Related procedures

The Generalized Linear Models procedure is a powerful tool for fitting a variety of models.

- The Generalized Estimating Equations procedure extends the generalized linear model to allow repeated measurements.
- The Linear Mixed Models procedure allows you to fit models for scale dependent variables with a random component and/or repeated measurements.

Recommended readings

See the following texts for more information on generalized linear models:

Cameron, A. C., and P. K. Trivedi. 1998. *Regression Analysis of Count Data*. Cambridge: Cambridge University Press. Dobson, A. J. 2002. *An Introduction to Generalized Linear Models*, 2 ed. Boca Raton, FL: Chapman & Hall/CRC. Hardin, J. W., and J. M. Hilbe. 2003. *Generalized Linear Models and Extension*. Station, TX: Stata Press. McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*, 2nd ed. London: Chapman & Hall.

Using Poisson regression to analyze ship damage rates (Generalized Linear Models)

A generalized linear model can be used to fit a Poisson regression for the analysis of count data. For example, a dataset presented and analyzed elsewhere¹ concerns damage to cargo ships caused by waves. The incident counts can be modeled as occurring at a Poisson rate given the values of the predictors, and the resulting model can help you determine which ship types are most prone to damage.

This example uses the stream *ships_genlin.str*, which references the data file *ships.sav*. The data file is in the *Demos* folder and the stream file is in the *streams* subfolder.

Modeling the raw cell counts can be misleading in this situation because the *Aggregate months of service* varies by ship type. Variables like this that measure the amount of "exposure" to risk are handled within the generalized linear model as offset variables. Moreover, a Poisson regression assumes that the log of the dependent variable is linear in the predictors. Thus, to use generalized linear models to fit a Poisson regression to the accident rates, you need to use *Logarithm of aggregate months of service*.

[Next](#)

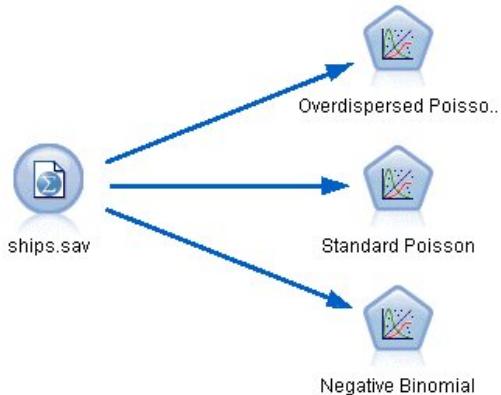
- [Fitting an "overdispersed" Poisson regression](#)
- [Goodness-of-fit statistics](#)
- [Omnibus test](#)
- [Tests of model effects](#)
- [Parameter estimates](#)
- [Fitting alternative models](#)
- [Goodness-of-fit statistics](#)
- [Summary](#)
- [Related procedures](#)
- [Recommended readings](#)

¹ McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*, 2nd ed. London: Chapman & Hall.

Fitting an "overdispersed" Poisson regression

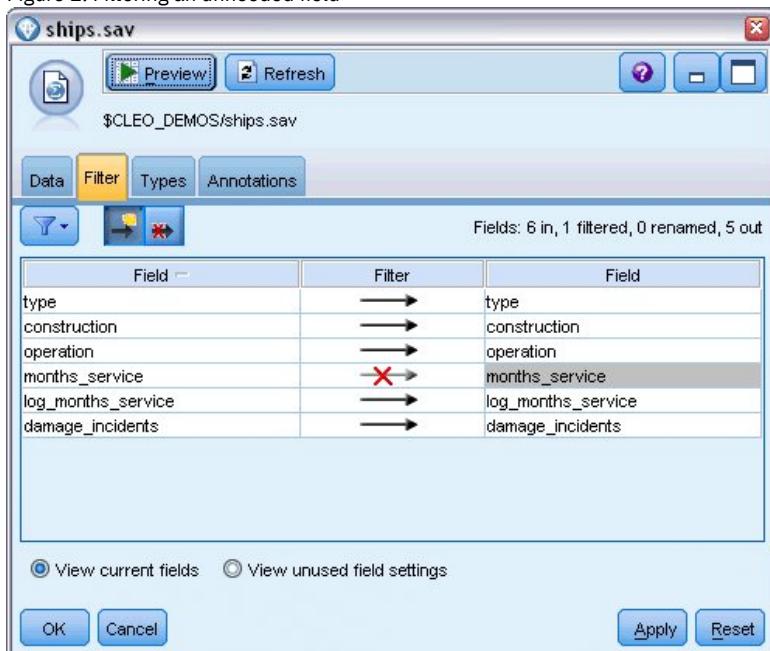
1. Add a Statistics File source node pointing to *ships.sav* in the *Demos* folder.

Figure 1. Sample stream to analyze damage rates



2. On the Filter tab of the source node, exclude the field *months_service*. The log-transformed values of this variable are contained in *log_months_service*, which will be used in the analysis.

Figure 2. Filtering an unneeded field



(Alternatively, you could change the role to None for this field on the Types tab rather than exclude it, or select the fields you want to use in the modeling node.)

3. On the Types tab of the source node, set the role for the *damage_incidents* field to Target. All other fields should have their role set to Input.
4. Click Read Values to instantiate the data.

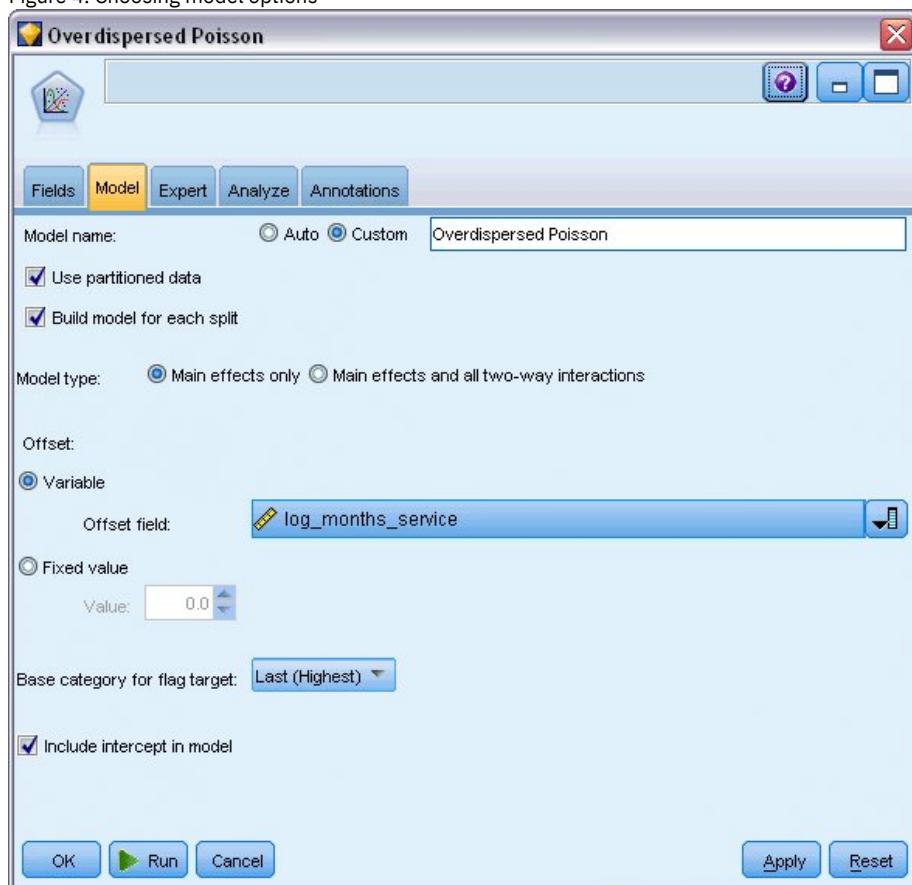
Figure 3. Setting field role



5. Attach a Genlin node to the source node; on the Genlin node, click the Model tab.

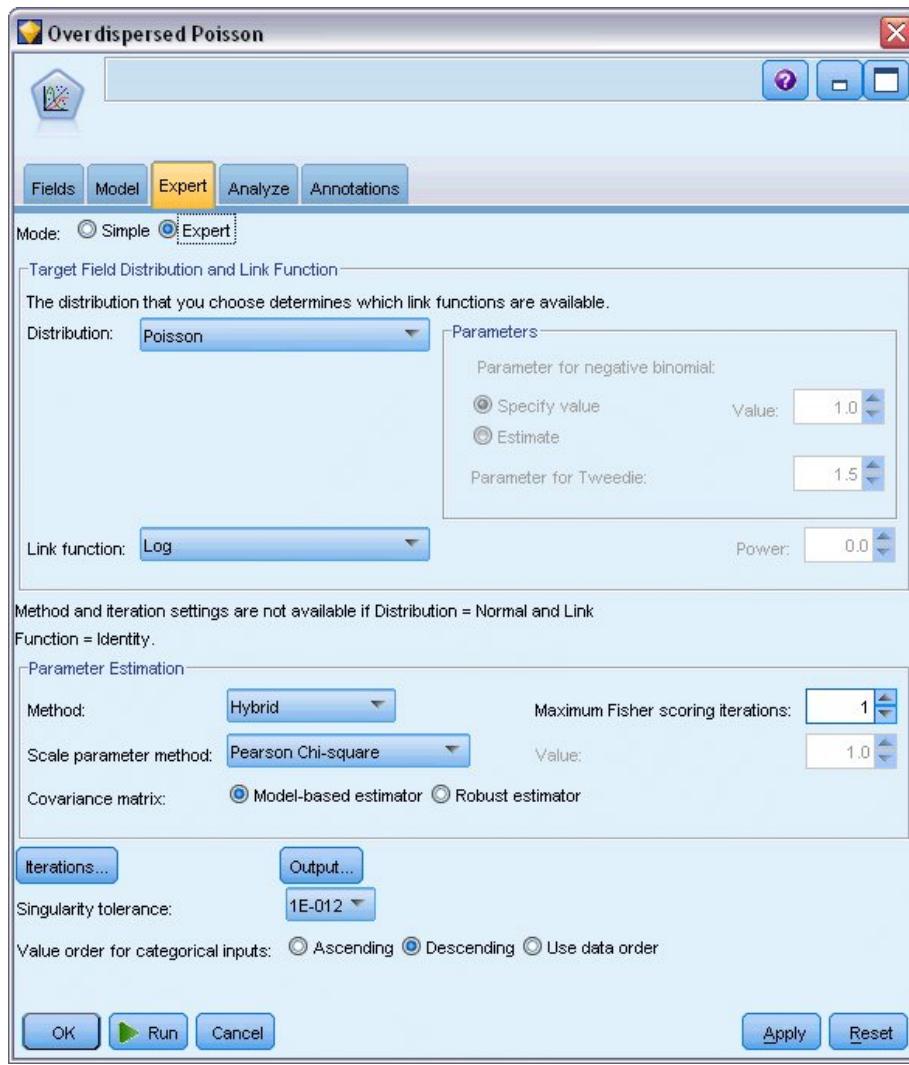
6. Select *log_months_service* as the offset variable.

Figure 4. Choosing model options



7. Click the Expert tab and select Expert to activate the expert modeling options.

Figure 5. Choosing expert options



8. Select Poisson as the distribution for the response and Log as the link function.
9. Select Pearson Chi-Square as the method for estimating the scale parameter. The scale parameter is usually assumed to be 1 in a Poisson regression, but McCullagh and Nelder use the Pearson chi-square estimate to obtain more conservative variance estimates and significance levels.
10. Select Descending as the category order for factors. This indicates that the first category of each factor will be its reference category; the effect of this selection on the model is in the interpretation of parameter estimates.
11. Click Run to create the model nugget, which is added to the stream canvas, and also to the Models palette in the upper right corner. To view the model details, right-click the nugget and choose Edit or Browse, then click the Advanced tab.

[Next](#)

Goodness-of-fit statistics

Figure 1. Goodness-of-fit statistics

	Value	df	Value/df
Deviance	38.695	25	1.548
Scaled Deviance	22.883	25	
Pearson Chi-Square	42.275	25	1.691
Scaled Pearson Chi-Square	25.000	25	
Log Likelihood ^a	-68.281		
Akaike's Information Criterion (AIC)	154.562		
Finite Sample Corrected AIC (AICC)	162.062		
Bayesian Information Criterion (BIC)	168.299		
Consistent AIC (CAIC)	177.299		

Dependent Variable: Number of damage incidents

Model: (Intercept), type, construction, operation, offset = log_months_service

- a. The full log likelihood function is displayed and used in computing information criteria.
- b. Information criteria are in small-is-better form.

The goodness-of-fit statistics table provides measures that are useful for comparing competing models. Additionally, the *Value/df* for the Deviance and Pearson Chi-Square statistics gives corresponding estimates for the scale parameter. These values should be near 1.0 for a Poisson regression; the fact that they are greater than 1.0 indicates that fitting the overdispersed model may be reasonable.

[Next](#)

Omnibus test

Figure 1. Omnibus test

Omnibus Test^a

Likelihood Ratio Chi-Square	df	Sig.
63.650	8	.000

Dependent Variable: Number of damage incidents

Model: (Intercept), Year of construction, Period of operation, Ship type, offset = Logarithm of aggregate months of service

- a. Compares the fitted model against the intercept-only model.

The omnibus test is a likelihood-ratio chi-square test of the current model versus the null (in this case, intercept) model. The significance value of less than 0.05 indicates that the current model outperforms the null model.

[Next](#)

Tests of model effects

Figure 1. Tests of model effects

Tests of Model Effects

Source	Type III		
	Wald Chi-Square	df	Sig.
(Intercept)	2138.657	1	.000
Year of construction	17.242	3	.001
Period of operation	6.249	1	.012
Ship type	15.415	4	.004

Dependent Variable: Number of damage incidents

Model: (Intercept), Year of construction, Period of operation, Ship type, offset = Logarithm of aggregate months of service

Each term in the model is tested for whether it has any effect. Terms with significance values less than 0.05 have some discernible effect. Each of the main-effects terms contributes to the model.

[Next](#)

Parameter estimates

Figure 1. Parameter estimates

Parameter Estimates

Parameter	B	Std. Error	95% Wald Confidence Interval		Hypothesis Test		
			Lower	Upper	Wald Chi-Square	df	Sig.
(Intercept)	-6.406	.2828	-6.960	-5.852	513.238	1	.000
[Year of construction=75]	.453	.3032	-.141	1.048	2.236	1	.135
[Year of construction=70]	.818	.2208	.386	1.251	13.743	1	.000
[Year of construction=65]	.697	.1946	.316	1.079	12.835	1	.000
[Year of construction=60]	0 ^a
[Period of operation=75]	.384	.1538	.083	.686	6.249	1	.012
[Period of operation=60]	0 ^a
[Ship type=5]	.326	.3067	-.276	.927	1.127	1	.288
[Ship type=4]	-.076	.3779	-.817	.665	.040	1	.841
[Ship type=3]	-.687	.4279	-1.526	.151	2.581	1	.108
[Ship type=2]	-.543	.2309	-.996	-.091	5.536	1	.019
[Ship type=1]	0 ^a
(Scale)	1.691 ^b						

Dependent Variable: Number of damage incidents

Model: (Intercept), Year of construction, Period of operation, Ship type, offset = Logarithm of aggregate months of service

a. Set to zero because this parameter is redundant.

b. Computed based on the Pearson chi-square.

The parameter estimates table summarizes the effect of each predictor. While interpretation of the coefficients in this model is difficult because of the nature of the link function, the signs of the coefficients for covariates and relative values of the coefficients for factor levels can give important insights into the effects of the predictors in the model.

- For covariates, positive (negative) coefficients indicate positive (inverse) relationships between predictors and outcome. An increasing value of a covariate with a positive coefficient corresponds to an increasing rate of damage incidents.
- For factors, a factor level with a greater coefficient indicates greater incidence of damage. The sign of a coefficient for a factor level is dependent upon that factor level's effect relative to the reference category.

You can make the following interpretations based on the parameter estimates:

- Ship type *B* [*type=2*] has a statistically significantly (*p* value of 0.019) lower damage rate (estimated coefficient of -0.543) than type *A* [*type=1*], the reference category. Type *C* [*type=3*] actually has an estimated parameter lower than *B*, but the variability in *C*'s estimate clouds the effect. See the estimated marginal means for all relations between factor levels.
- Ships constructed between 1965–69 [*construction=65*] and 1970–74 [*construction=70*] have statistically significantly (*p* values <0.001) higher damage rates (estimated coefficients of 0.697 and 0.818, respectively) than those built between 1960–64 [*construction=60*], the reference category. See the estimated marginal means for all relations between factor levels.
- Ships in operation between 1975–79 [*operation=75*] have statistically significantly (*p* value of 0.012) higher damage rates (estimated coefficient of 0.384) than those in operation between 1960–1974 [*operation=60*].

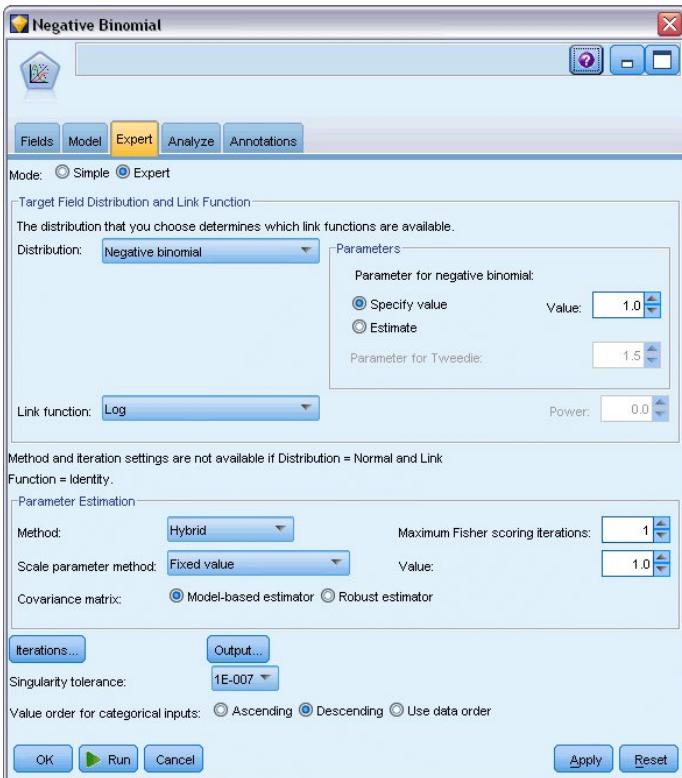
[Next](#)

Fitting alternative models

One problem with the "overdispersed" Poisson regression is that there is no formal way to test it versus the "standard" Poisson regression. However, one suggested formal test to determine whether there is overdispersion is to perform a likelihood ratio test between a "standard" Poisson regression and a negative binomial regression with all other settings equal. If there is no overdispersion in the Poisson regression, then the statistic $-2 \times (\text{log-likelihood for Poisson model} - \text{log-likelihood for negative binomial model})$ should have a mixture distribution with half its probability mass at 0 and the rest in a chi-square distribution with 1 degree of freedom.

- Select Fixed value as the method for estimating the scale parameter. By default, this value is 1.

Figure 1. Expert tab



2. To fit the negative binomial regression, copy and paste the Genlin node, attach it to the source node, open the new node and click the Expert tab.
3. Select Negative binomial as the distribution. Leave the default value of 1 for the ancillary parameter.
4. Run the stream and browse the Advanced tab on the newly-created model nuggets.

[Next](#)

Goodness-of-fit statistics

Figure 1. Goodness-of-fit statistics for standard Poisson regression

	Value	df	Value/df
Deviance	38.695	25	1.548
Scaled Deviance	38.695	25	
Pearson Chi-Square	42.275	25	1.691
Scaled Pearson Chi-Square	42.275	25	
Log Likelihood ^a	-68.281		
Akaike's Information Criterion (AIC)	154.562		
Finite Sample Corrected AIC (AICC)	162.062		
Bayesian Information Criterion (BIC)	168.299		
Consistent AIC (CAIC)	177.299		

Dependent Variable: Number of damage incidents
Model: (Intercept), type, construction, operation, offset = log_months_service

- a. The full log likelihood function is displayed and used in computing information criteria.
- b. Information criteria are in small-is-better form.

The log-likelihood reported for the standard Poisson regression is -68.281. Compare this to the negative binomial model.

Figure 2. Goodness-of-fit statistics for negative binomial regression

	Value	df	Value/df
Deviance	11.145	25	.446
Scaled Deviance	11.145	25	
Pearson Chi-Square	8.815	25	.353
Scaled Pearson Chi-Square	8.815	25	
Log Likelihood ^a	-83.725		
Akaike's Information Criterion (AIC)	185.450		
Finite Sample Corrected AIC (AICC)	192.950		
Bayesian Information Criterion (BIC)	199.187		
Consistent AIC (CAIC)	208.187		

Dependent Variable: Number of damage incidents
Model: (Intercept), type, construction, operation, offset = log_months_service

- a. The full log likelihood function is displayed and used in computing information criteria.
- b. Information criteria are in small-is-better form.

The log-likelihood reported for the negative binomial regression is -83.725. This is actually *smaller* than the log-likelihood for the Poisson regression, which indicates (without the need for a likelihood ratio test) that this negative binomial regression does not offer an improvement over the Poisson regression.

However, the chosen value of 1 for the ancillary parameter of the negative binomial distribution may not be optimal for this dataset. Another way you could test for overdispersion is to fit a negative binomial model with ancillary parameter equal to 0 and request the Lagrange multiplier test on the Output dialog of the Expert tab. If the test is not significant, overdispersion should not be a problem for this dataset.

[Next](#)

Summary

Using Generalized Linear Models, you have fit three different models for count data. The negative binomial regression was shown not to offer any improvement over the Poisson regression. The overdispersed Poisson regression seems to offer a reasonable alternative to the standard Poisson model, but there is not a formal test for choosing between them.

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the *IBM SPSS Modeler Algorithms Guide*.

Related procedures

The Generalized Linear Models procedure is a powerful tool for fitting a variety of models.

- The Generalized Estimating Equations procedure extends the generalized linear model to allow repeated measurements.
- The Linear Mixed Models procedure allows you to fit models for scale dependent variables with a random component and/or repeated measurements.

Recommended readings

See the following texts for more information on generalized linear models:

Cameron, A. C., and P. K. Trivedi. 1998. *Regression Analysis of Count Data*. Cambridge: Cambridge University Press. Dobson, A. J. 2002. *An Introduction to Generalized Linear Models*, 2 ed. Boca Raton, FL: Chapman & Hall/CRC. Hardin, J. W., and J. M. Hilbe. 2003. *Generalized Linear Models and Extension*. Station, TX: Stata Press. McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*, 2nd ed. London: Chapman & Hall.

Fitting a Gamma regression to car insurance claims (Generalized Linear Models)

A generalized linear model can be used to fit a Gamma regression for the analysis of positive range data. For example, a dataset presented and analyzed elsewhere [1](#) concerns damage claims for cars. The average claim amount can be modeled as having a gamma distribution, using an inverse link function to relate the mean of the dependent variable to a linear combination of the predictors. In order to account for the varying number of claims used to compute the average claim amounts, you specify *Number of claims* as the scaling weight.

This example uses the stream named *car-insurance_genlin.str*, which references the data file named *car_insurance_claims.sav*. The data file is in the *Demos* folder and the stream file is in the *streams* subfolder.

[Next](#)

- [Creating the Stream](#)
- [Parameter estimates](#)
- [Summary](#)
- [Related procedures](#)
- [Recommended readings](#)

¹ McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*, 2nd ed. London: Chapman & Hall.

Creating the Stream

1. Add a Statistics File source node pointing to *car_insurance_claims.sav* in the *Demos* folder.

Figure 1. Sample stream to predict car insurance claims



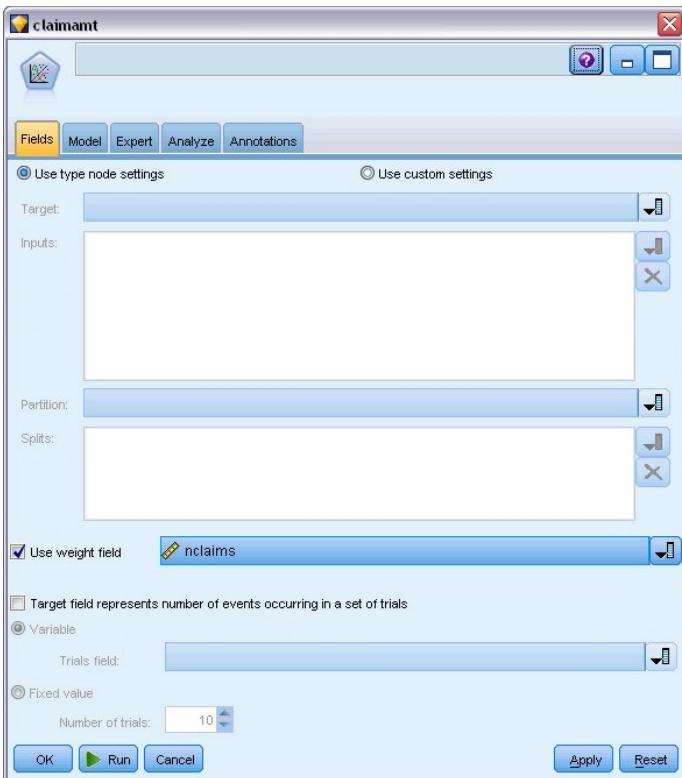
2. On the Types tab of the source node, set the role for the *claimamt* field to Target. All other fields should have their role set to Input.
3. Click Read Values to instantiate the data.

Figure 2. Setting field role



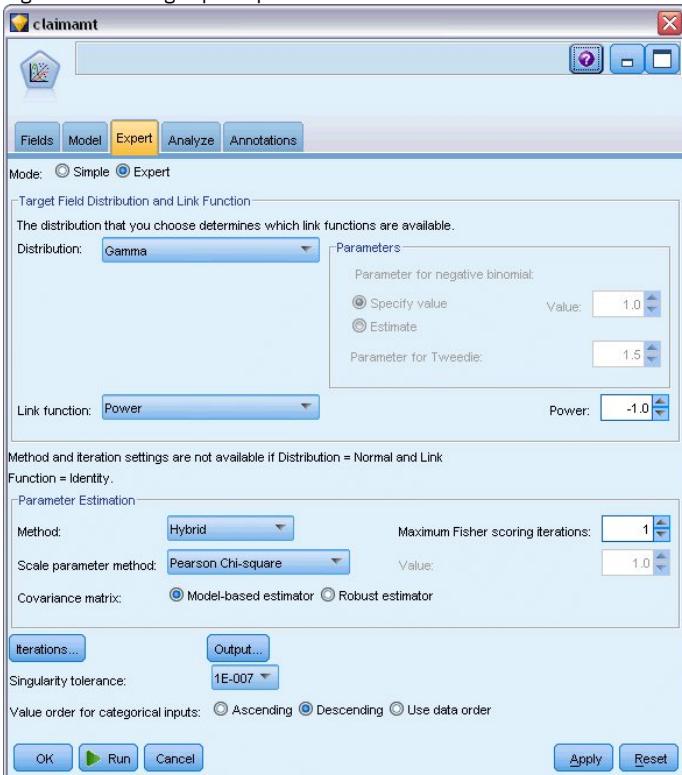
4. Attach a Genlin node to the source node; in the Genlin node, click the Fields tab.
5. Select *nclaims* as the scale weight field.

Figure 3. Choosing field options



6. Click the Expert tab and select Expert to activate the expert modeling options.

Figure 4. Choosing expert options



7. Select Gamma as the response distribution.
8. Select Power as the link function and type -1.0 as the exponent of the power function. This is an inverse link.
9. Select Pearson chi-square as the method for estimating the scale parameter. This is the method used by McCullagh and Nelder, so we follow it here in order to replicate their results.
10. Select Descending as the category order for factors. This indicates that the first category of each factor will be its reference category; the effect of this selection on the model is in the interpretation of parameter estimates.
11. Click Run to create the model nugget, which is added to the stream canvas, and also to the Models palette in the upper-right corner. To view the model details, right-click the model nugget and choose Edit or Browse, then select the Advanced tab.

Parameter estimates

Figure 1. Parameter estimates

Parameter Estimates

Parameter	B	Std. Error	95% Wald Confidence Interval		Hypothesis Test		
			Lower	Upper	Wald Chi-Square	df	Sig.
(Intercept)	.003	.0004	.003	.004	66.593	1	.000
[Policyholder age=8]	.001	.0004	.000	.002	4.898	1	.027
[Policyholder age=7]	.001	.0004	.000	.002	5.046	1	.025
[Policyholder age=6]	.001	.0004	.000	.002	5.740	1	.017
[Policyholder age=5]	.001	.0004	.001	.002	10.682	1	.001
[Policyholder age=4]	.000	.0004	.000	.001	1.268	1	.260
[Policyholder age=3]	.000	.0004	.000	.001	.720	1	.396
[Policyholder age=2]	.000	.0004	-.001	.001	.054	1	.816
[Policyholder age=1]	0 ^a
[Vehicle age=4]	.004	.0004	.003	.005	88.175	1	.000
[Vehicle age=3]	.002	.0002	.001	.002	53.013	1	.000
[Vehicle age=2]	.000	.0001	.000	.001	13.191	1	.000
[Vehicle age=1]	0 ^a
[Vehicle group=4]	-.001	.0002	-.002	-.001	61.883	1	.000
[Vehicle group=3]	-.001	.0002	-.001	.000	13.039	1	.000
[Vehicle group=2]	3.765E-5	.0002	.000	.000	.050	1	.823
[Vehicle group=1]	0 ^a
(Scale)	1.209 ^b						

Dependent Variable: Average cost of claims

Model: (Intercept), Policyholder age, Vehicle age, Vehicle group

a. Set to zero because this parameter is redundant.

b. Computed based on the Pearson chi-square.

The omnibus test and tests of model effects (not shown) indicate that the model outperforms the null model and that each of the main effects terms contribute to the model. The parameter estimates table shows the same values obtained by McCullagh and Nelder for the factor levels and the scale parameter.

Summary

Using Generalized Linear Models, you have fit a gamma regression to the claims data. Note that while the canonical link function for the gamma distribution was used in this model, a log link will also give reasonable results. In general, it is difficult to impossible to directly compare models with different link functions; however, the log link is a special case of the power link where the exponent is 0, thus you can compare the deviances of a model with a log link and a model with a power link to determine which gives the better fit (see, for example, section 11.3 of McCullagh and Nelder).

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the *IBM SPSS Modeler Algorithms Guide*.

Related procedures

The Generalized Linear Models procedure is a powerful tool for fitting a variety of models.

- The Generalized Estimating Equations procedure extends the generalized linear model to allow repeated measurements.
- The Linear Mixed Models procedure allows you to fit models for scale dependent variables with a random component and/or repeated measurements.

Recommended readings

See the following texts for more information on generalized linear models:

Cameron, A. C., and P. K. Trivedi. 1998. *Regression Analysis of Count Data*. Cambridge: Cambridge University Press. Dobson, A. J. 2002. *An Introduction to Generalized Linear Models*, 2 ed. Boca Raton, FL: Chapman & Hall/CRC. Hardin, J. W., and J. M. Hilbe. 2003. *Generalized Linear Models and Extension*. Station, TX: Stata Press. McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*, 2nd ed. London: Chapman & Hall.

Classifying Cell Samples (SVM)

Support Vector Machine (SVM) is a classification and regression technique that is particularly suitable for wide datasets. A wide dataset is one with a large number of predictors, such as might be encountered in the field of bioinformatics (the application of information technology to biochemical and biological data).

A medical researcher has obtained a dataset containing characteristics of a number of human cell samples extracted from patients who were believed to be at risk of developing cancer. Analysis of the original data showed that many of the characteristics differed significantly between benign and malignant samples. The researcher wants to develop an SVM model that can use the values of these cell characteristics in samples from other patients to give an early indication of whether their samples might be benign or malignant.

This example uses the stream named `svm_cancer.str`, available in the `Demos` folder under the `streams` subfolder. The data file is `cell_samples.data`. See the topic [Demos Folder](#) for more information.

The example is based on a dataset that is publicly available from the UCI Machine Learning Repository . The dataset consists of several hundred human cell sample records, each of which contains the values of a set of cell characteristics. The fields in each record are:

Field name	Description
<code>ID</code>	Patient identifier
<code>Clump</code>	Clump thickness
<code>UnifSize</code>	Uniformity of cell size
<code>UnifShape</code>	Uniformity of cell shape
<code>MargAdh</code>	Marginal adhesion
<code>SingEpiSize</code>	Single epithelial cell size
<code>BareNuc</code>	Bare nuclei
<code>BlandChrom</code>	Bland chromatin
<code>NormNucl</code>	Normal nucleoli
<code>Mit</code>	Mitoses
<code>Class</code>	Benign or malignant

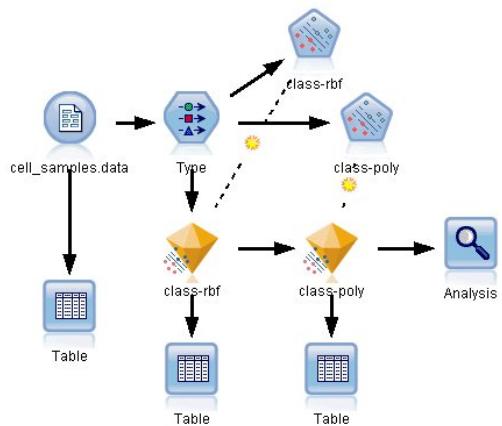
For the purposes of this example, we're using a dataset that has a relatively small number of predictors in each record.

[Next](#)

- [Creating the Stream](#)
- [Examining the Data](#)
- [Trying a Different Function](#)
- [Comparing the Results](#)
- [Summary](#)

Creating the Stream

Figure 1. Sample stream to show SVM modeling



1. Create a new stream and add a Var File source node pointing to *cell_samples.data* in the Demos folder of your IBM® SPSS® Modeler installation.

Let's take a look at the data in the source file.

2. Add a Table node to the stream.
3. Attach the Table node to the Var File node and run the stream.

Figure 2. Source data for SVM

The screenshot shows the 'Table' dialog box with the title 'Table (11 fields, 699 records)'. The table view displays the first 20 rows of the dataset. The columns are labeled: ID, hilSize, UnifShape, MargAdh, SingEpiSize, BareNuc, BlandChrom, NormNuci, Mit, and Class. The data shows various numerical values for each patient sample, with the 'Class' column indicating the final diagnosis.

ID	hilSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNuci	Mit	Class
1	1	1	2	1	3	1	1	2	
2	4	5	7	10	3	2	1	2	
3	1	1	2	2	3	1	1	2	
4	8	1	3	4	3	7	1	2	
5	1	3	2	1	3	1	1	2	
6	10	8	7	10	9	7	1	4	
7	1	1	2	10	3	1	1	2	
8	2	1	2	1	3	1	1	2	
9	1	1	2	1	1	1	5	2	
10	1	1	2	1	2	1	1	2	
11	1	1	1	1	3	1	1	2	
12	1	1	2	1	2	1	1	2	
13	3	3	2	3	4	4	1	4	
14	1	1	2	3	3	1	1	2	
15	5	10	7	9	5	5	4	4	
16	6	4	6	1	4	3	1	4	
17	1	1	2	1	2	1	1	2	
18	1	1	2	1	3	1	1	2	
19	7	6	4	10	4	1	2	4	
20	1	1	2	1	3	1	1	2	

The *ID* field contains the patient identifiers. The characteristics of the cell samples from each patient are contained in fields *Clump* to *Mit*. The values are graded from 1 to 10, with 1 being the closest to benign.

The *Class* field contains the diagnosis, as confirmed by separate medical procedures, as to whether the samples are benign (value = 2) or malignant (value = 4).

Figure 3. Type node settings



4. Add a Type node and attach it to the Var File node.

5. Open the Type node.

We want the model to predict the value of *Class* (that is, benign (=2) or malignant (=4)). As this field can have one of only two possible values, we need to change its measurement level to reflect this.

6. In the Measurement column for the *Class* field (the last one in the list), click the value Continuous and change it to Flag.

7. Click Read Values.

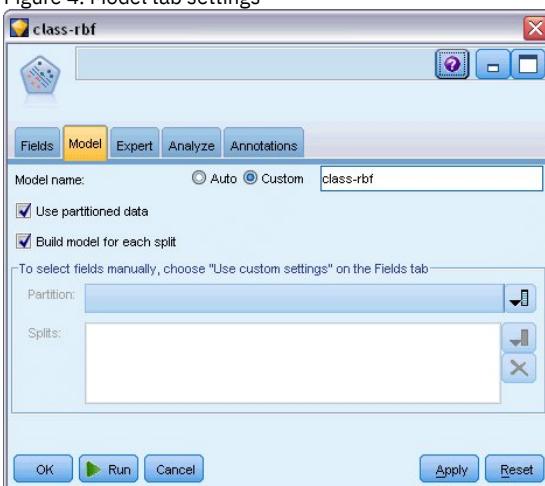
8. In the Role column, set the role for *ID* (the patient identifier) to None, as this will not be used either as a predictor or a target for the model.

9. Set the role for the target, *Class*, to Target and leave the role of all the other fields (the predictors) as Input.

10. Click OK.

The SVM node offers a choice of kernel functions for performing its processing. As there's no easy way of knowing which function performs best with any given dataset, we'll choose different functions in turn and compare the results. Let's start with the default, RBF (Radial Basis Function).

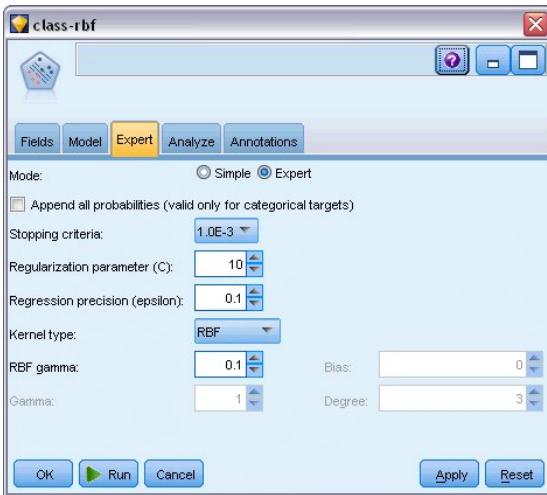
Figure 4. Model tab settings



11. From the Modeling palette, attach an SVM node to the Type node.

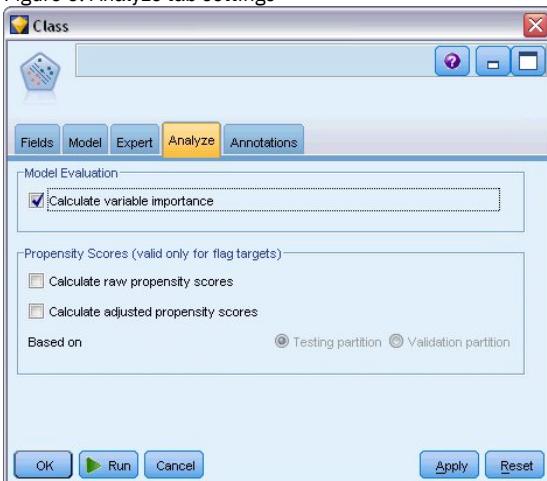
12. Open the SVM node. On the Model tab, click the Custom option for Model name and type *class-rbf* in the adjacent text field.

Figure 5. Default Expert tab settings



13. On the Expert tab, set the Mode to Expert for readability but leave all the default options as they are. Note that Kernel type is set to RBF by default. All the options are greyed out in Simple mode.

Figure 6. Analyze tab settings

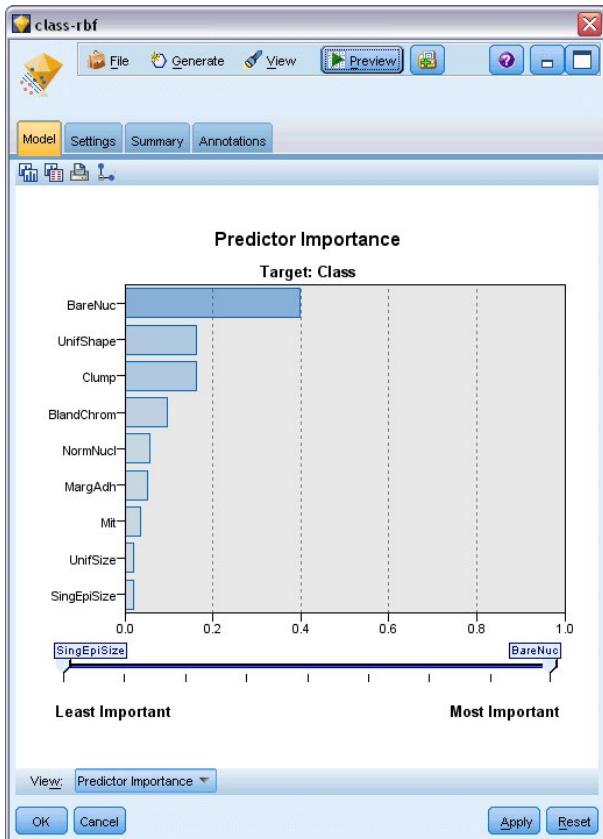


14. On the Analyze tab, select the Calculate variable importance check box.
 15. Click Run. The model nugget is placed in the stream, and in the Models palette at the top right of the screen.
 16. Double-click the model nugget in the stream.

[Next](#)

Examining the Data

Figure 1. Predictor Importance graph



On the Model tab, the Predictor Importance graph shows the relative effect of the various fields on the prediction. This shows us that *BareNuc* has easily the greatest effect, while *UnifShape* and *Clump* are also quite significant.

1. Click OK.
2. Attach a Table node to the *class-rbf* model nugget.
3. Open the Table node and click Run.

Figure 2. Fields added for prediction and confidence value

The table output shows the following data structure:

	gEpiSize	BareNuc	BlandChrom	NormNuc	Mit	Class	\$S-Class	\$SP-Class
1		1	3	1	1	2	2	0.992
2		10	3	2	1	2	4	0.899
3		2	3	1	1	2	2	0.994
4		4	3	7	1	2	4	0.915
5		1	3	1	1	2	2	0.992
6		10	9	7	1	4	4	0.999
7		10	3	1	1	2	2	0.907
8		1	3	1	1	2	2	0.997
9		1	1	1	5	2	2	0.997
10		1	2	1	1	2	2	0.996
11		1	3	1	1	2	2	0.999
12		1	2	1	1	2	2	0.999
13		3	4	4	1	4	2	0.514
14		3	3	1	1	2	2	0.989
15		9	5	5	4	4	4	0.991
16		1	4	3	1	4	4	0.691
17		1	2	1	1	2	2	0.997
18		1	3	1	1	2	2	0.995
19		10	4	1	2	4	4	0.996
20		1	3	1	1	2	2	0.986

4. The model has created two extra fields. Scroll the table output to the right to see them:

New field name	Description
\$S-Class	Value for <i>Class</i> predicted by the model.
\$SP-Class	Propensity score for this prediction (the likelihood of this prediction being true, a value from 0.0 to 1.0).

Just by looking at the table, we can see that the propensity scores (in the *\$SP-Class* column) for most of the records are reasonably high.

However, there are some significant exceptions; for example, the record for patient 1041801 at line 13, where the value of 0.514 is unacceptably low. Also, comparing *Class* with *SS-Class*, it's clear that this model has made a number of incorrect predictions, even where the propensity score was relatively high (for example, lines 2 and 4).

Let's see if we can do better by choosing a different function type.

[Next](#)

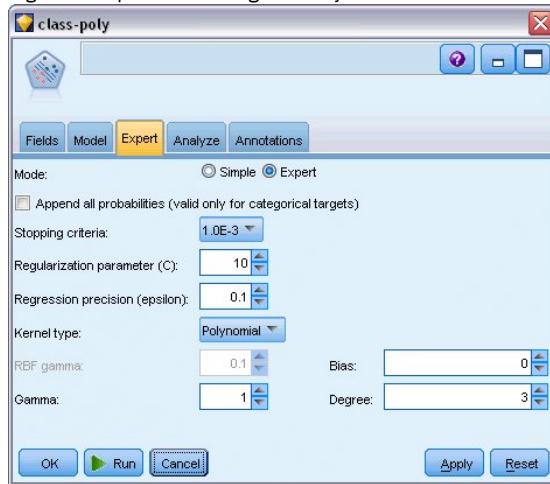
Trying a Different Function

Figure 1. Setting a new name for the model



1. Close the Table output window.
2. Attach a second SVM modeling node to the Type node.
3. Open the new SVM node.
4. On the Model tab, choose Custom and type *class-poly* as the model name.

Figure 2. Expert tab settings for Polynomial



5. On the Expert tab, set Mode to Expert.
6. Set Kernel type to Polynomial and click Run. The *class-poly* model nugget is added to the stream, and also to the Models palette at the top right of the screen.
7. Connect the *class-rbf* model nugget to the *class-poly* model nugget (choose Replace at the warning dialog).
8. Attach a Table node to the *class-poly* nugget.
9. Open the Table node and click Run.

[Next](#)

Comparing the Results

Figure 1. Fields added for Polynomial function

The screenshot shows a 'Table' node window with 15 fields and 699 records. The columns include 'ormNucl', 'Mit', 'Class', '\$S-Class', '\$SP-Class', '\$S1-Class', and '\$SP1-Class'. The '\$S1-Class' and '\$SP1-Class' columns contain values ranging from 0.992 to 1.000.

	ormNucl	Mit	Class	\$S-Class	\$SP-Class	\$S1-Class	\$SP1-Class
78		1	2	2	0.992	2	0.998
79		1	2	2	0.968	2	0.967
80		1	2	2	0.998	2	0.994
81		1	2	2	0.986	2	0.991
82		1	2	2	0.996	2	0.997
83		1	2	2	0.991	2	0.998
84		1	2	2	0.970	2	0.998
85	0	7	4	4	0.992	4	1.000
86	0	10	4	4	0.974	4	1.000
87		1	4	4	0.786	4	0.958
88		3	4	4	0.988	4	0.935
89		1	2	2	0.995	2	0.997
90		1	2	2	0.998	2	0.991
91		1	2	2	0.999	2	0.993
92		1	2	2	0.998	2	0.996
93		1	2	2	0.995	2	0.997
94		1	2	2	0.999	2	0.994
95		1	2	2	0.998	2	0.995
96		1	2	2	0.999	2	0.993
97		1	2	2	0.999	2	0.995

1. Scroll the table output to the right to see the newly added fields.

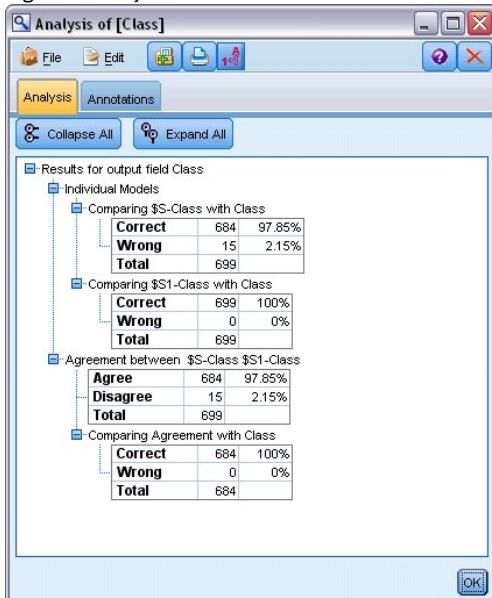
The generated fields for the Polynomial function type are named **\$S1-Class** and **\$SP1-Class**.

The results for Polynomial look much better. Many of the propensity scores are 0.995 or better, which is very encouraging.

2. To confirm the improvement in the model, attach an Analysis node to the *class-poly* model nugget.

Open the Analysis node and click Run.

Figure 2. Analysis node



This technique with the Analysis node enables you to compare two or more model nuggets of the same type. The output from the Analysis node shows that the RBF function correctly predicts 97.85% of the cases, which is still quite good. However, the output shows that the Polynomial function has correctly predicted the diagnosis in every single case. In practice you are unlikely to see 100% accuracy, but you can use the Analysis node to help determine whether the model is acceptably accurate for your particular application.

In fact, neither of the other function types (Sigmoid and Linear) performs as well as Polynomial on this particular dataset. However, with a different dataset, the results could easily be different, so it's always worth trying the full range of options.

[Next](#)

Summary

You have used different types of SVM kernel functions to predict a classification from a number of attributes. You have seen how different kernels give different results for the same dataset and how you can measure the improvement of one model over another.

Using Cox Regression to Model Customer Time to Churn

As part of its efforts to reduce customer churn, a telecommunications company is interested in modeling the "time to churn" in order to determine the factors that are associated with customers who are quick to switch to another service. To this end, a random sample of customers is selected and their time spent as customers, whether they are still active customers, and various other fields are pulled from the database.

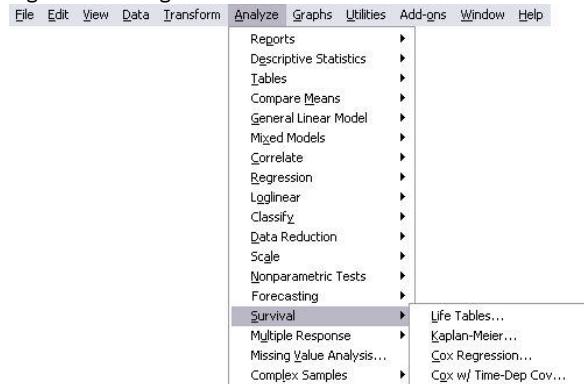
This example uses the stream *telco_coxreg.str*, which references the data file *telco.sav*. The data file is in the *Demos* folder and the stream file is in the *streams* subfolder. See the topic [Demos Folder](#) for more information.

[Next](#)

- [Building a Suitable Model](#)
- [Tracking the Expected Number of Customers Retained](#)
- [Scoring](#)
- [Summary](#)

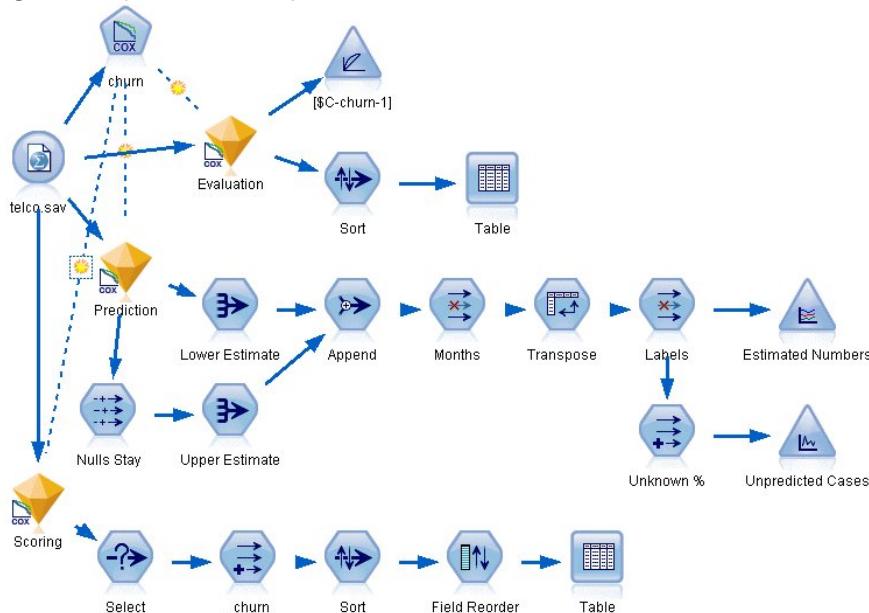
Building a Suitable Model

Figure 1. Cox Regression menu selection



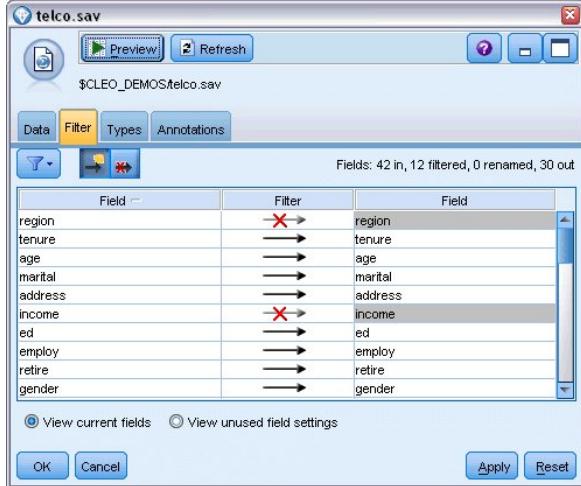
1. Add a Statistics File source node pointing to *telco.sav* in the *Demos* folder.

Figure 2. Sample stream to analyze time to churn



2. On the Filter tab of the source node, exclude the fields *region*, *income*, *longten* through *wireten*, and *loglong* through *logwire*.

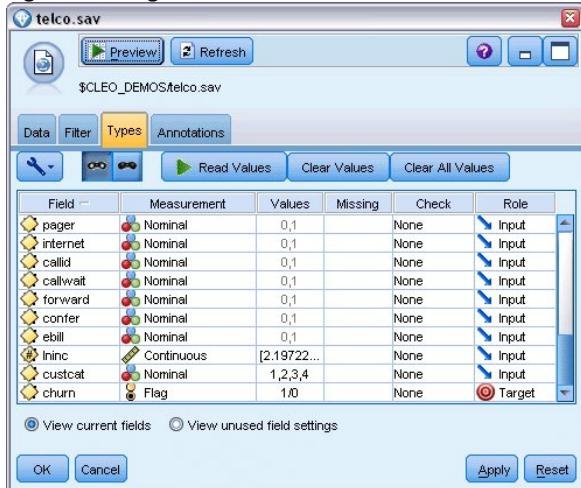
Figure 3. Filtering unneeded fields



(Alternatively, you could change the role to None for these fields on the Types tab rather than exclude it, or select the fields you want to use in the modeling node.)

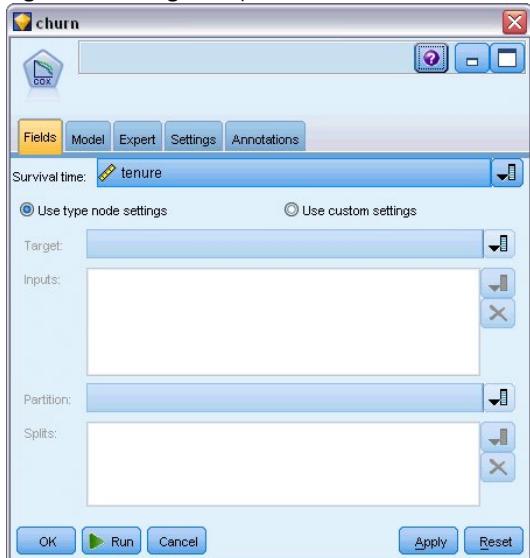
3. On the Types tab of the source node, set the role for the *churn* field to Target and set its measurement level to Flag. All other fields should have their role set to Input.
4. Click Read Values to instantiate the data.

Figure 4. Setting field role



5. Attach a Cox node to the source node; in the Fields tab, select *tenure* as the survival time variable.

Figure 5. Choosing field options



6. Click the Model tab.
7. Select Stepwise as the variable selection method.

Figure 6. Choosing model options



8. Click the Expert tab and select Expert to activate the expert modeling options.
9. Click Output.

Figure 7. Choosing advanced output options



10. Select Survival and Hazard as plots to produce, then click OK.
11. Click Run to create the model nugget, which is added to the stream, and to the Models palette in the upper right corner. To view its details, double-click the nugget on the stream. First, look at the Advanced output tab.

[Next](#)

[Next](#)

- [Censored Cases](#)
- [Categorical Variable Codings](#)
- [Variable Selection](#)
- [Covariate Means](#)
- [Survival Curve](#)
- [Hazard Curve](#)
- [Evaluation](#)

Censored Cases

Figure 1. Case processing summary

		N	Percent
Cases available in analysis	Event ^a	274	27.4%
	Censored	726	72.6%
	Total	1000	100.0%
Cases dropped	Cases with missing values	0	0.0%
	Cases with negative time	0	0.0%
	Censored cases before the earliest event in a stratum	0	0.0%
	Total	0	0.0%
Total		1000	100.0%

a. Dependent Variable: Months with service

The status variable identifies whether the event has occurred for a given case. If the event has not occurred, the case is said to be censored. Censored cases are not used in the computation of the regression coefficients but are used to compute the baseline hazard. The case processing summary shows that 726 cases are censored. These are customers who have not churned.

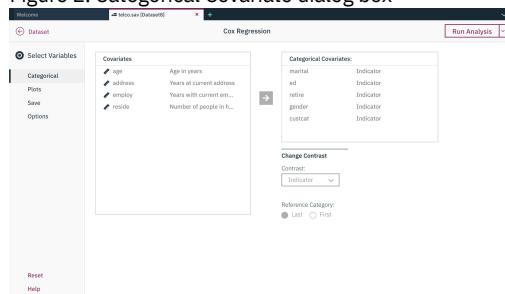
[Next](#)

Categorical Variable Codings

Figure 1. Categorical variable codings

		Frequency	(1) ^b	(2)	(3)	(4)
marital ^a	0=Unmarried	505	1			
	1=Married	495	0			
ed ^a	1=Did not complete high school	204	1	0	0	0
	2=High school degree	287	0	1	0	0
	3=Some college	209	0	0	1	0
	4=College degree	234	0	0	0	1
	5=Post-undergraduate degree	66	0	0	0	0
	.00=No	953	1			
retire ^a	1.00=Yes	47	0			
	gender ^a	483	1			
gender ^a	1=Female	517	0			
	tollfree ^a	526	1			
tollfree ^a	1=Yes	474	0			
	equip ^a	614	1			
equip ^a	1=Yes	386	0			
	callcard ^a	322	1			
callcard ^a	1=Yes	678	0			
	wireless ^a	704	1			
wireless ^a	1=Yes	296	0			
	multiline ^a	525	1			
multiline ^a	1=Yes	475	0			
	voice ^a	696	1			
voice ^a	1=Yes	304	0			
	pager ^a	739	1			
pager ^a	1=Yes	261	0			
	internet ^a	632	1			
internet ^a	1=Yes	368	0			
	callid ^a	519	1			
callid ^a	1=Yes	481	0			
	callwait ^a	515	1			
callwait ^a	1=Yes	485	0			
	forward ^a	507	1			
forward ^a	1=Yes	493	0			
	confer ^a	498	1			
confer ^a	1=Yes	502	0			
	ebill ^a	629	1			
ebill ^a	1=Yes	371	0			
	custcat ^a	266	1	0	0	
custcat ^a	2=E-service	217	0	1	0	
	3=Plus service	281	0	0	1	
	4=Total service	236	0	0	0	

Figure 2. Categorical Covariate dialog box



The categorical variable codings are a useful reference for interpreting the regression coefficients for categorical covariates, particularly dichotomous variables. By default, the reference category is the "last" category. Thus, for example, even though *Married* customers have variable values of 1 in the data file, they are coded as 0 for the purposes of the regression.

[Next](#)

Variable Selection

Figure 1. Omnibus tests

Step	-2 Log Likelihood	Overall (score)			Change From Previous Step			Change From Previous Block		
		Chi-square	df	Sig.	Chi-square	df	Sig.	Chi-square	df	Sig.
1 ^a	3392.536	162.303	1	.000	133.828	1	.000	133.828	1	.000
2 ^b	3087.314	249.392	2	.000	305.222	1	.000	439.050	2	.000
3 ^c	3027.085	328.426	3	.000	60.229	1	.000	499.279	3	.000
4 ^d	2990.790	347.197	4	.000	36.294	1	.000	535.574	4	.000
5 ^e	2973.790	362.673	5	.000	17.000	1	.000	552.574	5	.000
6 ^f	2958.796	376.140	6	.000	14.994	1	.000	567.568	6	.000
7 ^g	2945.503	384.717	7	.000	13.293	1	.000	580.861	7	.000
8 ^h	2936.993	417.341	8	.000	8.510	1	.004	589.371	8	.000
9 ⁱ	2926.000	423.911	9	.000	10.994	1	.001	600.364	9	.000
10 ^j	2917.551	428.078	10	.000	8.449	1	.004	608.813	10	.000
11 ^k	2913.308	436.837	11	.000	4.243	1	.039	613.056	11	.000
12 ^l	2908.078	440.158	12	.000	5.230	1	.022	618.286	12	.000

- a. Variable(s) Entered at Step Number 1: callcard
 - b. Variable(s) Entered at Step Number 2: longmon
 - c. Variable(s) Entered at Step Number 3: equip
 - d. Variable(s) Entered at Step Number 4: employ
 - e. Variable(s) Entered at Step Number 5: multiline
 - f. Variable(s) Entered at Step Number 6: voice
 - g. Variable(s) Entered at Step Number 7: address
 - h. Variable(s) Entered at Step Number 8: equipmon
 - i. Variable(s) Entered at Step Number 9: ebill
 - j. Variable(s) Entered at Step Number 10: callid
 - k. Variable(s) Entered at Step Number 11: internet
 - l. Variable(s) Entered at Step Number 12: reside
- m. Beginning Block Number 0, Initial Log Likelihood function: -2 Log likelihood: 3526.364
n. Beginning Block Number 1, Method = Forward Stepwise (Likelihood Ratio)

The model-building process employs a forward stepwise algorithm. The omnibus tests are measures of how well the model performs. The chi-square change from previous step is the difference between the -2 log-likelihood of the model at the previous step and the current step. If the step was to add a variable, the inclusion makes sense if the significance of the change is less than 0.05. If the step was to remove a variable, the exclusion makes sense if the significance of the change is greater than 0.10. In twelve steps, twelve variables are added to the model.

Figure 2. Variables in the equation (step 12 only)

Step 12		B	SE	Wald	df	Sig.	Exp(B)
		address	-.035	.009	14.543	1	.000
	employ	-.051	.010	25.767	1	.000	.950
	reside	-.103	.046	5.037	1	.025	.902
	equip	-1.948	.381	26.180	1	.000	.143
	callcard	.777	.151	26.451	1	.000	2.175
	longmon	-.233	.022	115.619	1	.000	.792
	equipmon	-.042	.011	15.377	1	.000	.859
	multiline	.612	.145	17.854	1	.000	1.844
	voice	.501	.157	10.197	1	.001	.606
	internet	-.362	.160	5.114	1	.024	.697
	callid	-.464	.148	9.790	1	.002	.629
	ebill	-.399	.156	6.557	1	.010	.671

The final model includes *address*, *employ*, *reside*, *equip*, *callcard*, *longmon*, *equipmon*, *multiline*, *voice*, *internet*, *callid*, and *ebill*. To understand the effects of individual predictors, look at *Exp(B)*, which can be interpreted as the predicted change in the hazard for a unit increase in the predictor.

- The value of *Exp(B)* for *address* means that the churn hazard is reduced by $100\% - (100\% \times 0.966) = 3.4\%$ for each year a customer has lived at the same address. The churn hazard for a customer who has lived at the same address for five years is reduced by $100\% - (100\% \times 0.966^5) = 15.88\%$.
- The value of *Exp(B)* for *callcard* means that the churn hazard for a customer who does not subscribe to the calling card service is 2.175 times that of a customer with the service. Recall from the categorical variable codings that *No* = 1 for the regression.
- The value of *Exp(B)* for *internet* means that the churn hazard for a customer who does not subscribe to the internet service is 0.697 times that of a customer with the service. This is somewhat worrisome because it suggests that customers with the service are leaving the company faster than customers without the service.

Figure 3. Variables not in the model (step 12 only)

		Score	df	Sig.
Step 12	age	.122	1	.726
	marital	.648	1	.421
	income	1.476	1	.224
	ed	6.328	4	.176
	ed(1)	.007	1	.934
	ed(2)	.203	1	.652
	ed(3)	.835	1	.361
	ed(4)	5.773	1	.016
	retire	.013	1	.908
	gender	.214	1	.644
	tollfree	3.243	1	.072
	wireless	.668	1	.414
	tollmon	.000	1	.987
	cardmon	3.163	1	.075
	wiremon	1.084	1	.298
	pager	1.808	1	.179
	callwait	.266	1	.606
	forward	2.201	1	.138
	confer	2.568	1	.109
	custcat	.864	3	.834
	custcat(1)	.466	1	.495
	custcat(2)	.450	1	.502
	custcat(3)	.019	1	.889

Variables left out of the model all have score statistics with significance values greater than 0.05. However, the significance values for *tollfree* and *cardmon*, while not less than 0.05, are fairly close. They may be interesting to pursue in further studies.

[Next](#)

Covariate Means

Figure 1. Covariate means

	Mean
age	41.684
marital	.505
address	11.551
income	77.535
ed(1)	.204
ed(2)	.287
ed(3)	.209
ed(4)	.234
employ	10.987
retire	.953
gender	.483
reside	2.331
tollfree	.526
equip	.814
callcard	.322
wireless	.704
longmon	11.723
tollmon	13.274
equipmon	14.220
cardmon	13.781
wiremon	11.584
multline	.525
voice	.896
pager	.739
internet	.632
callid	.519
callwait	.515
forward	.507
confer	.498
ebill	.629
custcat(1)	.266
custcat(2)	.217
custcat(3)	.281

Figure 2. Choosing advanced output options

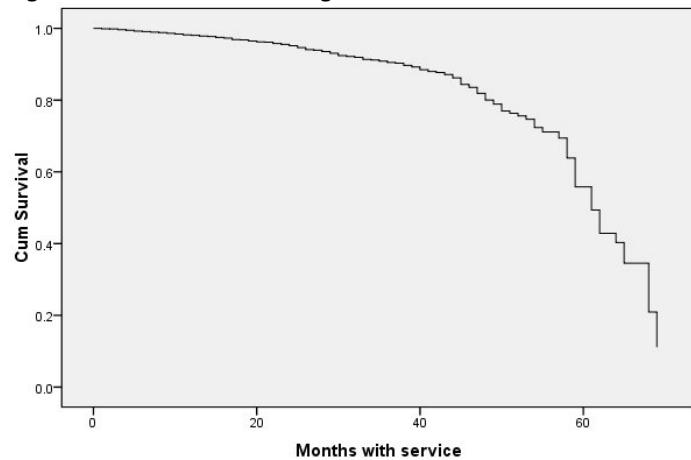


This table displays the average value of each predictor variable. This table is a useful reference when looking at the survival plots, which are constructed for the mean values. Note, however, that the "average" customer doesn't actually exist when you look at the means of indicator variables for categorical predictors. Even with all scale predictors, you are unlikely to find a customer whose covariate values are all close to the mean. If you want to see the survival curve for a particular case, you can change the covariate values at which the survival curve is plotted in the Plots dialog box. If you want to see the survival curve for a particular case, you can change the covariate values at which the survival curve is plotted in the Plots group of the Advanced Output dialog.

[Next](#)

Survival Curve

Figure 1. Survival curve for "average" customer

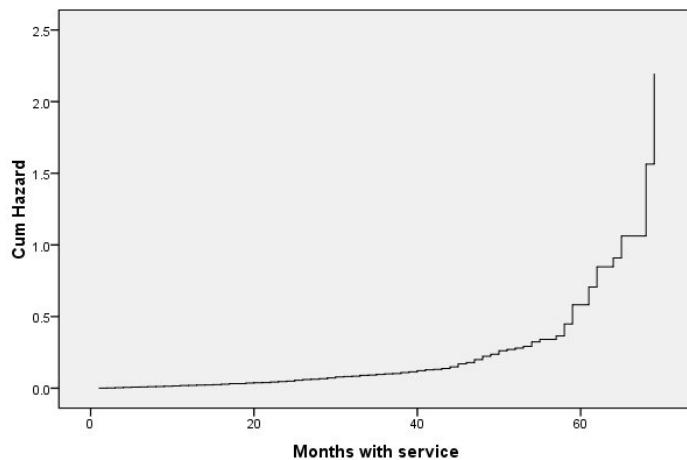


The basic survival curve is a visual display of the model-predicted time to churn for the "average" customer. The horizontal axis shows the time to event. The vertical axis shows the probability of survival. Thus, any point on the survival curve shows the probability that the "average" customer will remain a customer past that time. Past 55 months, the survival curve becomes less smooth. There are fewer customers who have been with the company for that long, so there is less information available, and thus the curve is blocky.

[Next](#)

Hazard Curve

Figure 1. Hazard curve for "average" customer



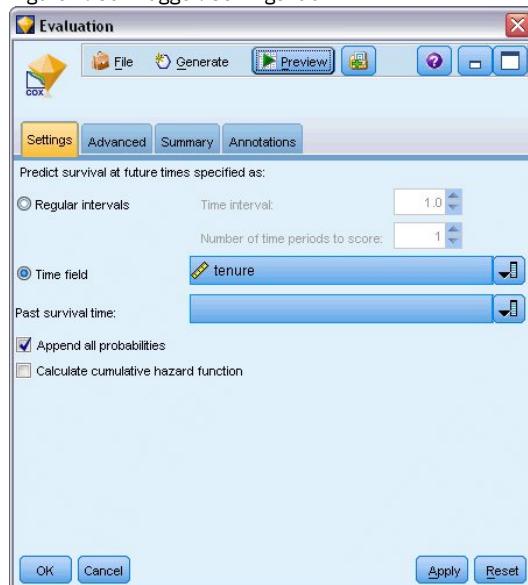
The basic hazard curve is a visual display of the cumulative model-predicted potential to churn for the "average" customer. The horizontal axis shows the time to event. The vertical axis shows the cumulative hazard, equal to the negative log of the survival probability. Past 55 months, the hazard curve, like the survival curve, becomes less smooth, for the same reason.

[Next](#)

Evaluation

The stepwise selection methods guarantee that your model will have only "statistically significant" predictors, but it does not guarantee that the model is actually good at predicting the target. To do this, you need to analyze scored records.

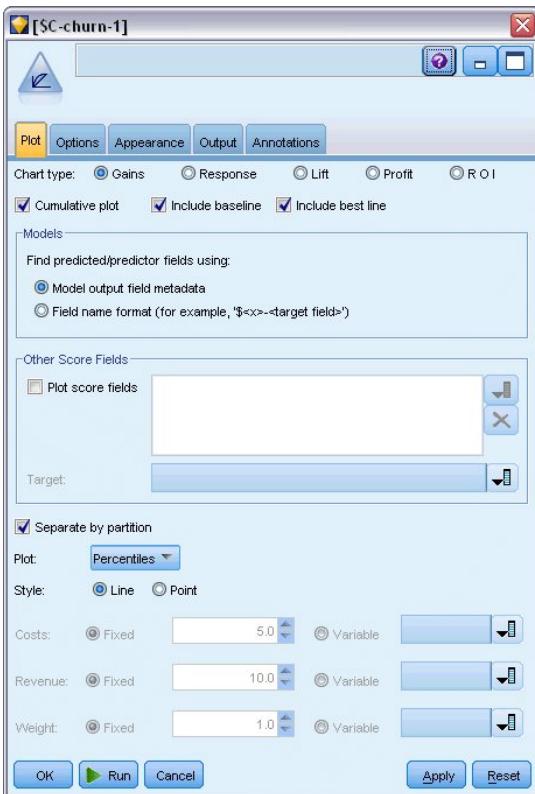
Figure 1. Cox nugget: Settings tab



1. Place the model nugget on the canvas and attach it to the source node, open the nugget and click the Settings tab.
2. Select Time field and specify *tenure*. Each record will be scored at its length of tenure.
3. Select Append all probabilities.

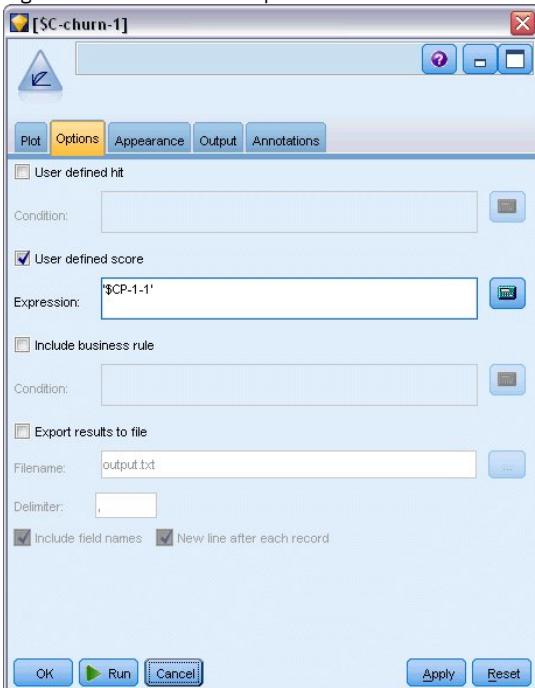
This creates scores using 0.5 as the cutoff for whether a customer churns; if their propensity to churn is greater than 0.5, they are scored as a churner. There is nothing magical about this number, and a different cutoff may yield more desirable results. For one way to think about choosing a cutoff, use an Evaluation node.

Figure 2. Evaluation node: Plot tab



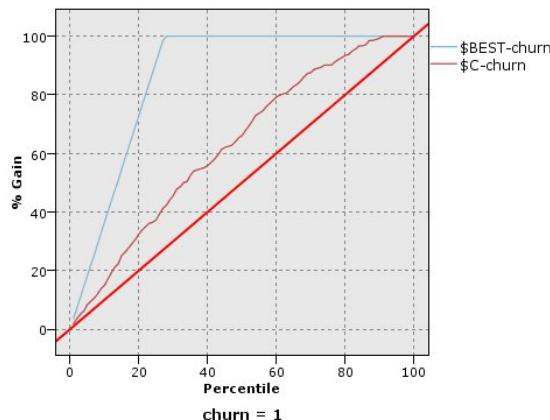
4. Attach an Evaluation node to the model nugget; on the Plot tab, select Include best line.
5. Click the Options tab.

Figure 3. Evaluation node: Options tab



6. Select User defined score and type '\$CP-1-1' as the expression. This is a model-generated field that corresponds to the propensity to churn.
7. Click Run.

Figure 4. Gains chart

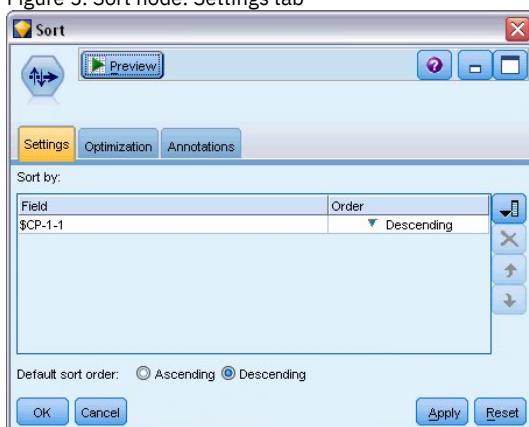


The cumulative gains chart shows the percentage of the overall number of cases in a given category "gained" by targeting a percentage of the total number of cases. For example, one point on the curve is at (10%, 15%), meaning that if you score a dataset with the model and sort all of the cases by predicted propensity to churn, you would expect the top 10% to contain approximately 15% of all of the cases that actually take the category 1 (churners). Likewise, the top 60% contains approximately 79.2% of the churners. If you select 100% of the scored dataset, you obtain all of the churners in the dataset.

The diagonal line is the "baseline" curve; if you select 20% of the records from the scored dataset at random, you would expect to "gain" approximately 20% of all of the records that actually take the category 1. The farther above the baseline a curve lies, the greater the gain. The "best" line shows the curve for a "perfect" model that assigns a higher churn propensity score to every chunner than every non-chunner. You can use the cumulative gains chart to help choose a classification cutoff by choosing a percentage that corresponds to a desirable gain, and then mapping that percentage to the appropriate cutoff value.

What constitutes a "desirable" gain depends on the cost of Type I and Type II errors. That is, what is the cost of classifying a chunner as a non-chunner (Type I)? What is the cost of classifying a non-chunner as a chunner (Type II)? If customer retention is the primary concern, then you want to lower your Type I error; on the cumulative gains chart, this might correspond to increased customer care for customers in the top 60% of predicted propensity of 1, which captures 79.2% of the possible churners but costs time and resources that could be spent acquiring new customers. If lowering the cost of maintaining your current customer base is the priority, then you want to lower your Type II error. On the chart, this might correspond to increased customer care for the top 20%, which captures 32.5% of the churners. Usually, both are important concerns, so you have to choose a decision rule for classifying customers that gives the best mix of sensitivity and specificity.

Figure 5. Sort node: Settings tab



8. Say that you have decided that 45.6% is a desirable gain, which corresponds to taking the top 30% of records. To find an appropriate classification cutoff, attach a Sort node to the model nugget.
9. On the Settings tab, choose to sort by **\$CP-1-1** in descending order and click OK.

Figure 6. Table

Table (34 fields, 1,000 records)

Irn	\$C-churn-1	\$CP-churn-1	\$CP-0-1	\$CP-1-1
292	0	0.744	0.744	0.256
293	0	0.745	0.745	0.255
294	0	0.745	0.745	0.255
295	0	0.746	0.746	0.254
296	0	0.748	0.748	0.252
297	0	0.749	0.749	0.251
298	0	0.749	0.749	0.251
299	0	0.750	0.750	0.250
300	0	0.752	0.752	0.248
301	0	0.752	0.752	0.248
302	0	0.754	0.754	0.246
303	0	0.754	0.754	0.246
304	0	0.755	0.755	0.245
305	0	0.756	0.756	0.244
306	0	0.757	0.757	0.243
307	0	0.757	0.757	0.243
308	0	0.758	0.758	0.242
309	0	0.759	0.759	0.241
310	0	0.761	0.761	0.239
311	0	0.762	0.762	0.238

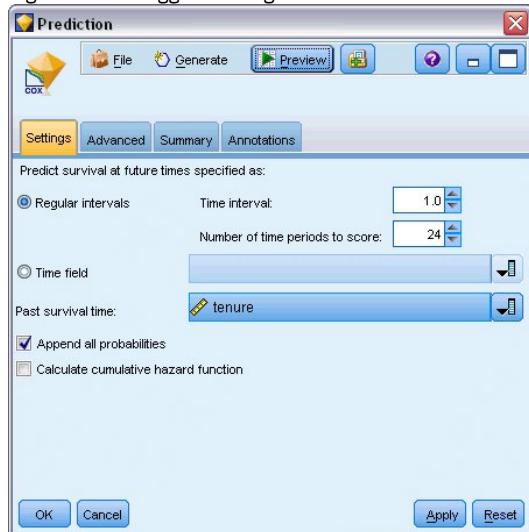
10. Attach a Table node to the Sort node.
11. Open the Table node and click Run.

Scrolling down the output, you see that the value of \$CP-1-1 is 0.248 for the 300th record. Using 0.248 as a classification cutoff should result in approximately 30% of the customers scored as churners, capturing approximately 45% of the actual total churners.

Tracking the Expected Number of Customers Retained

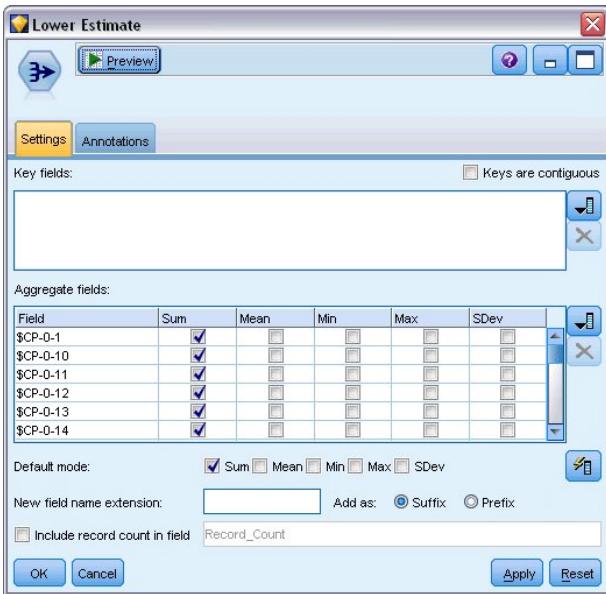
Once satisfied with a model, you want to track the expected number of customers in the dataset that are retained over the next two years. The null values, which are customers whose total tenure (future time + tenure) falls beyond the range of survival times in the data used to train the model, present an interesting challenge. One way to deal with them is to create two sets of predictions, one in which null values are assumed to have churned, and another in which they are assumed to have been retained. In this way you can establish upper and lower bounds on the expected number of customers retained.

Figure 1. Cox nugget: Settings tab



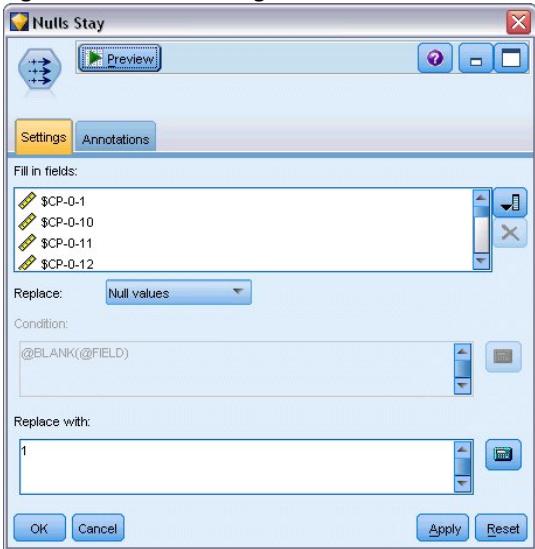
1. Double-click the model nugget in the Models palette (or copy and paste the nugget on the stream canvas) and attach the new nugget to the Source node.
2. Open the nugget to the Settings tab.
3. Make sure Regular Intervals is selected, and specify 1.0 as the time interval and 24 as the number of periods to score. This specifies that each record will be scored for each of the following 24 months.
4. Select *tenure* as the field to specify the past survival time. The scoring algorithm will take into account the length of each customer's time as a customer of the company.
5. Select Append all probabilities.

Figure 2. Aggregate node: Settings tab



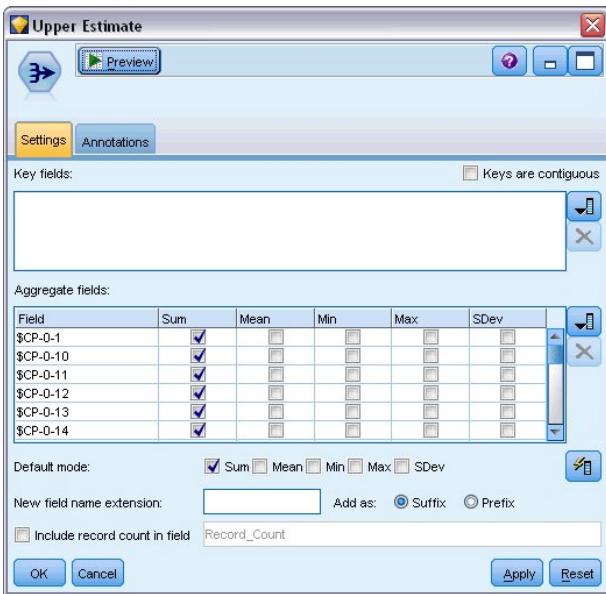
6. Attach an Aggregate node to the model nugget; on the Settings tab, deselect Mean as a default mode.
7. Select \$CP-0-1 through \$CP-0-24, the fields of form \$CP-0-n, as the fields to aggregate. This is easiest if, on the Select Fields dialog, you sort the fields by Name (that is, alphabetical order).
8. Deselect Include record count in field.
9. Click OK. This node creates the "lower bound" predictions.

Figure 3. Filler node: Settings tab



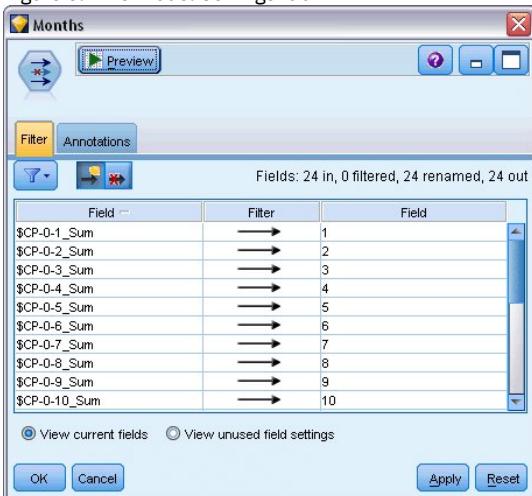
10. Attach a Filler node to the Coxreg nugget to which we just attached the Aggregate node; on the Settings tab, select \$CP-0-1 through \$CP-0-24, the fields of form \$CP-0-n, as the fields to fill in. This is easiest if, on the Select Fields dialog, you sort the fields by Name (that is, alphabetical order).
11. Choose to replace Null values with the value 1.
12. Click OK.

Figure 4. Aggregate node: Settings tab



13. Attach an Aggregate node to the Filler node; on the Settings tab, deselect Mean as a default mode.
14. Select \$CP-0-1 through \$CP-0-24, the fields of form \$CP-0-n, as the fields to aggregate. This is easiest if, on the Select Fields dialog, you sort the fields by Name (that is, alphabetical order).
15. Deselect Include record count in field.
16. Click OK. This node creates the "upper bound" predictions.

Figure 5. Filter node: Settings tab



17. Attach an Append node to the two Aggregate nodes, then attach a Filter node to the Append node.
18. On the Settings tab of the Filter node, rename the fields to 1 through 24. Through the use of a Transpose node, these field names will become values for the x-axis in charts downstream.

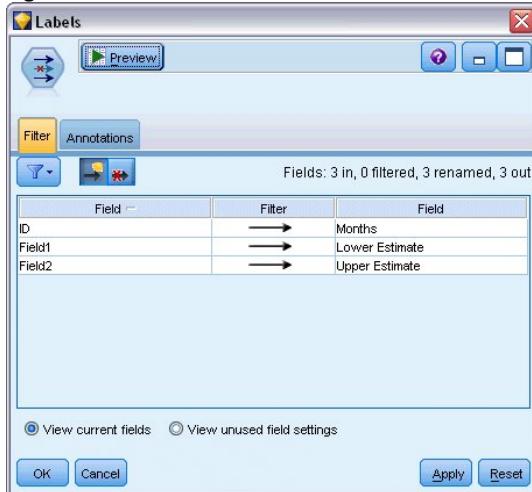
Figure 6. Transpose node: Settings tab



19. Attach a Transpose node to the Filter node.

20. Type 2 as the number of new fields.

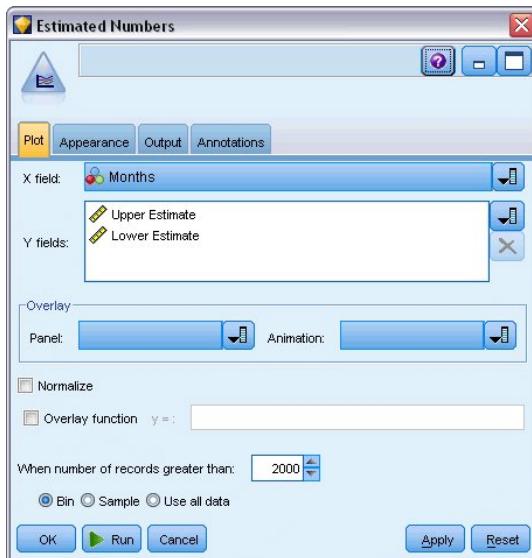
Figure 7. Filter node: Filter tab



21. Attach a Filter node to the Transpose node.

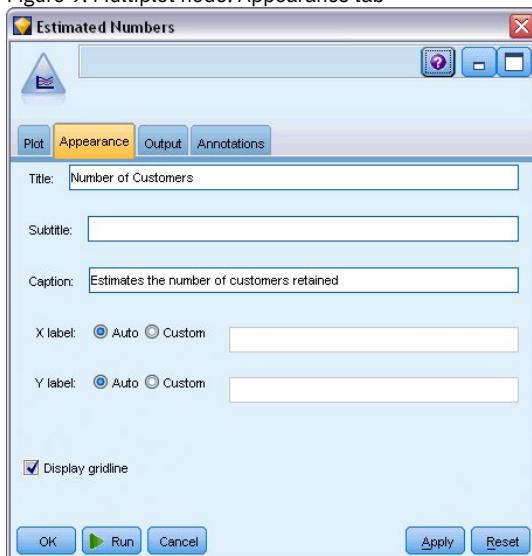
22. On the Settings tab of the Filter node, rename *ID* to *Months*, *Field1* to *Lower Estimate*, and *Field2* to *Upper Estimate*.

Figure 8. Multiplot node: Plot tab



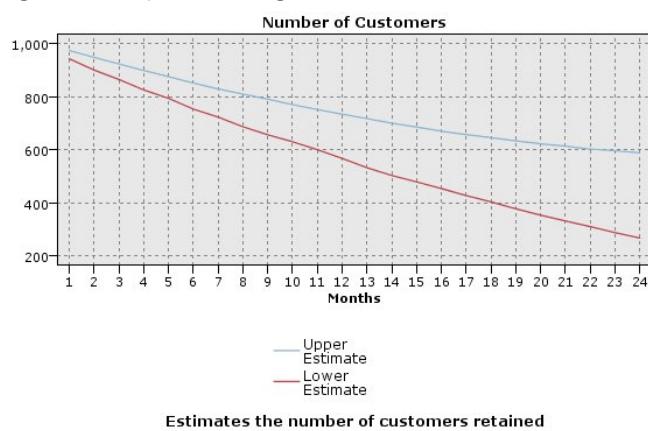
23. Attach a Multiplot node to the Filter node.
 24. On the Plot tab, *Months* as the X field, *Lower Estimate* and *Upper Estimate* as the Y fields.

Figure 9. Multiplot node: Appearance tab



25. Click the Appearance tab.
 26. Type *Number of Customers* as the title.
 27. Type *Estimates the number of customers retained* as the caption.
 28. Click Run.

Figure 10. Multiplot estimating the number of customers retained



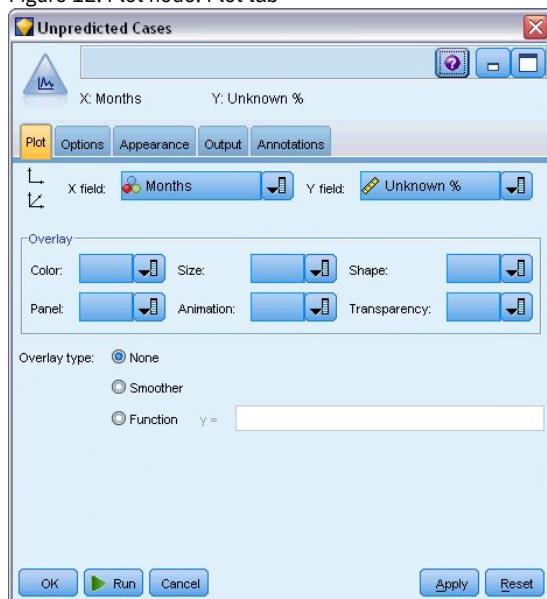
The upper and lower bounds on the estimated number of customers retained are plotted. The difference between the two lines is the number of customers scored as null, and therefore whose status is highly uncertain. Over time, the number of these customers increases. After 12 months, you can expect to retain between 601 and 735 of the original customers in the dataset; after 24 months, between 288 and 597.

Figure 11. Derive node: Settings tab



29. To get another look at how uncertain the estimates of the number of customers retained are, attach a Derive node to the Filter node.
30. On the Settings tab of the Derive node, type *Unknown %* as the derive field.
31. Select Continuous as the field type.
32. Type $(100 * ('Upper Estimate' - 'Lower Estimate')) / 'Lower Estimate'$ as the formula. *Unknown %* is the number of customers "in doubt" as a percentage of the lower estimate.
33. Click OK.

Figure 12. Plot node: Plot tab



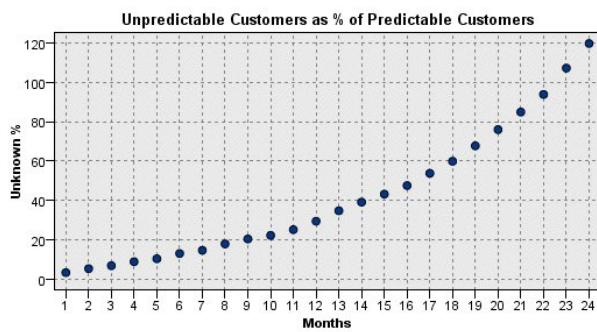
34. Attach a Plot node to the Derive node.
35. On the Plot tab of the Plot node, select *Months* as the X field and *Unknown %* as the Y field.
36. Click the Appearance tab.

Figure 13. Plot node: Appearance tab



37. Type Unpredictable Customers as % of Predictable Customers as the title.
 38. Execute the node.

Figure 14. Plot of unpredictable customers



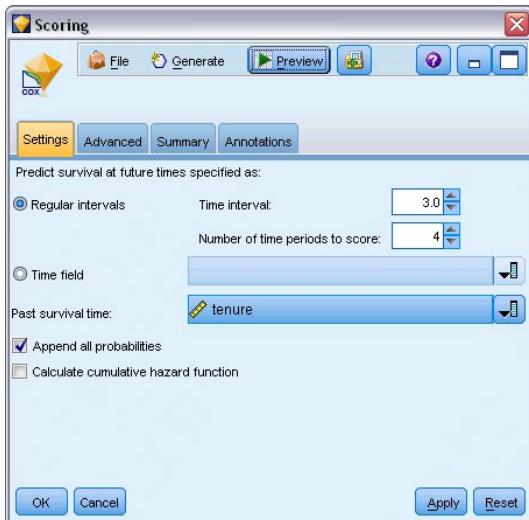
Through the first year, the percentage of unpredictable customers increases at a fairly linear rate, but the rate of increase explodes during the second year until, by month 23, the number of customers with null values outnumber the expected number of customers retained.

[Next](#)

Scoring

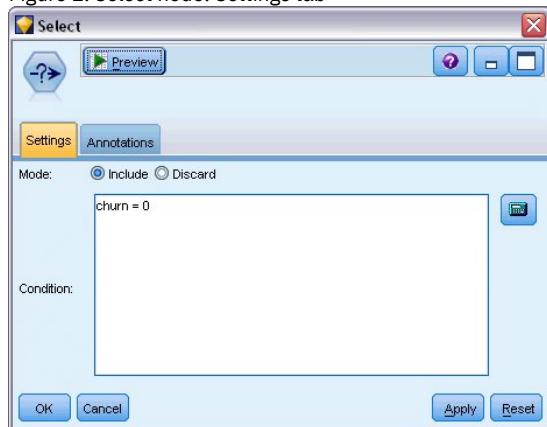
Once satisfied with a model, you want to score customers to identify the individuals most likely to churn within the next year, by quarter.

Figure 1. Coxreg nugget: Settings tab



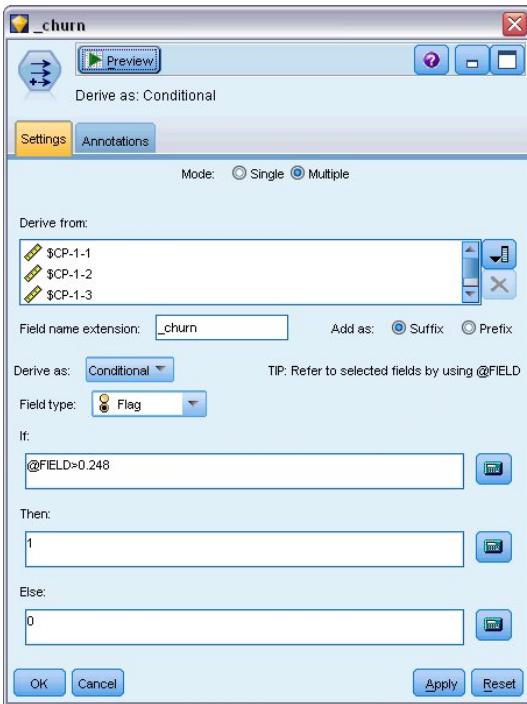
1. Attach a third model nugget to the Source node and open the model nugget.
2. Make sure Regular Intervals is selected, and specify 3.0 as the time interval and 4 as the number of periods to score. This specifies that each record will be scored for the following four quarters.
3. Select *tenure* as the field to specify the past survival time. The scoring algorithm will take into account the length of each customer's time as a customer of the company.
4. Select Append all probabilities. These extra fields will make it easier to sort the records for viewing in a table.

Figure 2. Select node: Settings tab



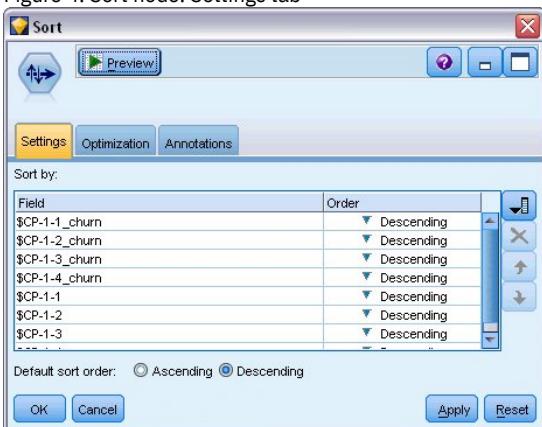
5. Attach a Select node to the model nugget; on the Settings tab, type `churn=0` as the condition. This removes customers who have already churned from the results table.

Figure 3. Derive node: Settings tab



6. Attach a Derive node to the Select node; on the Settings tab, select Multiple as the mode.
7. Choose to derive from \$CP-1-1 through \$CP-1-4, the fields of form \$CP-1-n, and type _churn as the suffix to add. This is easiest if, on the Select Fields dialog, you sort the fields by Name (that is, alphabetical order).
8. Choose to derive the field as a Conditional.
9. Select Flag as the measurement level.
10. Type @FIELD>0.248 as the If condition. Recall that this was the classification cutoff identified during Evaluation.
11. Type 1 as the Then expression.
12. Type 0 as the Else expression.
13. Click OK.

Figure 4. Sort node: Settings tab



14. Attach a Sort node to the Derive node; on the Settings tab, choose to sort by \$CP-1-1_churn through \$CP-1-4-churn and then \$CP-1-1 through \$CP-1-4, all in descending order. Customers who are predicted to churn will appear at the top.

Figure 5. Field Reorder node: Reorder tab



15. Attach a Field Reorder node to the Sort node; on the Reorder tab, choose to place `$CP-1-1_churn` through `$CP-1-4` in front of the other fields. This simply makes the results table easier to read, and so is optional. You will need to use the buttons to move the fields into the position shown in the figure.

Figure 6. Table showing customer scores

	\$CP-1-1_churn	\$CP-1-1	\$CP-1-2_churn	\$CP-1-2	\$CP-1-3_churn	\$CP-1-3	\$CP-1-4_churn	\$CP-1-4	tenur
255	0	0.032	0	0.075	0	0.147	1	0.298	49
256	0	0.027	0	0.064	0	0.127	1	0.260	49
257	0	0.023	0	0.130	0	0.233	1	0.308	53
258	0	0.021	0	0.127	0	0.239	1	0.320	54
259	0	0.021	0	0.125	0	0.237	1	0.318	54
260	0	0.021	0	0.053	0	0.198	1	0.331	50
261	0	0.021	0	0.053	0	0.196	1	0.329	50
262	0	0.020	0	0.050	0	0.189	1	0.317	50
263	0	0.017	0	0.043	0	0.163	1	0.278	50
264	0	0.015	0	0.039	0	0.148	1	0.253	50
265	0	0.197	0	0.197	0	\$null\$	0	\$null\$	66
266	0	0.109	0	0.109	0	\$null\$	0	\$null\$	66
267	0	0.101	0	0.214	0	\$null\$	0	\$null\$	65
268	0	0.081	0	0.137	0	0.194	0	0.245	23
269	0	0.074	0	0.159	0	\$null\$	0	\$null\$	65
270	0	0.070	0	0.116	0	0.158	0	0.237	28
271	0	0.070	0	0.128	0	0.189	0	0.234	45
272	0	0.062	0	0.105	0	0.151	0	0.191	23
273	0	0.062	0	0.130	0	0.163	0	0.212	44
274	0	0.061	0	0.123	0	0.182	0	0.241	4

16. Attach a Table node to the Field Reorder node and execute it.

264 customers are expected to churn by the end of the year, 184 by the end of the third quarter, 103 by the second, and 31 in the first. Note that given two customers, the one with a higher propensity to churn in the first quarter does not necessarily have a higher propensity to churn in later quarters; for example, see records 256 and 260. This is likely due to the shape of the hazard function for the months following the customer's current tenure; for example, customers who joined because of a promotion might be more likely to switch early on than customers who joined because of a personal recommendation, but if they do not then they may actually be more loyal for their remaining tenure. You may want to re-sort the customers to obtain different views of the customers most likely to churn.

Figure 7. Table showing customers with null values

Table (50 fields, 726 records)

At the bottom of the table are customers with predicted null values. These are customers whose total tenure (future time + *tenure*) falls beyond the range of survival times in the data used to train the model.

[Next](#)

Summary

Using Cox regression, you have found an acceptable model for the time to churn, plotted the expected number of customers retained over the next two years, and identified the individual customers most likely to churn in the next year. Note that while this is an acceptable model, it may not be the best model. Ideally you should at least compare this model, obtained using the Forward stepwise method, with one created using the Backward stepwise method.

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the *IBM SPSS Modeler Algorithms Guide*.

Market Basket Analysis (Rule Induction/C5.0)

This example deals with fictitious data describing the contents of supermarket baskets (that is, collections of items bought together) plus the associated personal data of the purchaser, which might be acquired through a loyalty card scheme. The goal is to discover groups of customers who buy similar products and can be characterized demographically, such as by age, income, and so on.

This example illustrates two phases of data mining:

- Association rule modeling and a web display revealing links between items purchased
- C5.0 rule induction profiling the purchasers of identified product groups

Note: This application does not make direct use of predictive modeling, so there is no accuracy measurement for the resulting models and no associated training/test distinction in the data mining process.

This example uses the stream named *baskrule*, which references the data file named *BASKETS1n*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *baskrule* file is in the *streams* directory.

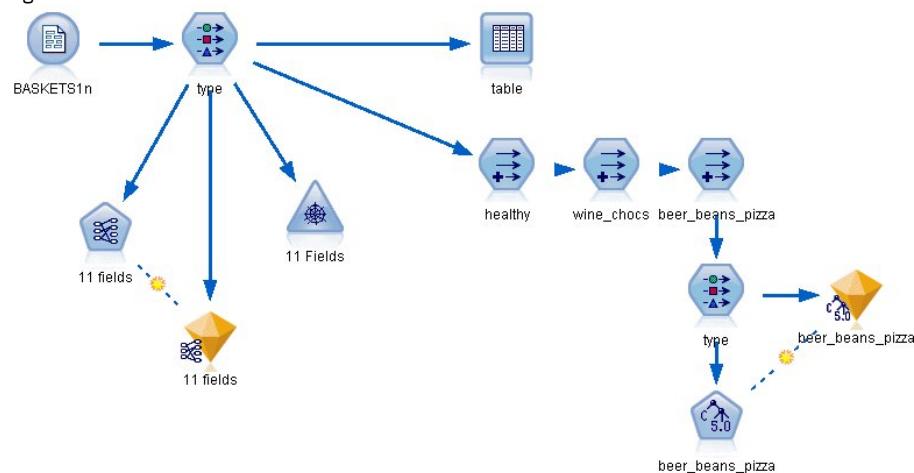
[Next](#)

- [Accessing the Data](#)
- [Discovering affinities in basket contents](#)
- [Profiling the Customer Groups](#)
- [Summary](#)

Accessing the Data

Using a Variable File node, connect to the dataset *BASKETS1n*, selecting to read field names from the file. Connect a Type node to the data source, and then connect the node to a Table node. Set the measurement level of the field *cardid* to *Typeless* (because each loyalty card ID occurs only once in the dataset and can therefore be of no use in modeling). Select *Nominal* as the measurement level for the field *sex* (this is to ensure that the Apriori modeling algorithm will not treat *sex* as a flag).

Figure 1. baskrule stream



Now run the stream to instantiate the Type node and display the table. The dataset contains 18 fields, with each record representing a basket.

The 18 fields are presented in the following headings.

Basket summary:

- *cardid*. Loyalty card identifier for customer purchasing this basket.
- *value*. Total purchase price of basket.
- *pmethod*. Method of payment for basket.

Personal details of cardholder:

- *sex*
- *homeown*. Whether or not cardholder is a homeowner.
- *income*
- *age*

Basket contents—flags for presence of product categories:

- *fruitveg*
- *freshmeat*
- *dairy*
- *cannedveg*
- *cannedmeat*
- *frozenmeal*
- *beer*
- *wine*
- *softdrink*
- *fish*
- *confectionery*

[Next](#)

Discovering affinities in basket contents

First, you need to acquire an overall picture of affinities (associations) in the basket contents using Apriori to produce association rules. Select the fields to be used in this modeling process by editing the Type node and setting the role of all of the product categories to *Both* and all other roles to *None*. (*Both* means that the field can be either an input or an output of the resultant model.)

Note: You can set options for multiple fields using Shift-click to select the fields before specifying an option from the columns.

Figure 1. Selecting fields for modeling



Once you have specified fields for modeling, attach an Apriori node to the Type node, edit it, select the option Only true values for flags, and click run on the Apriori node. The result, a model on the Models tab at the upper right of the managers window, contains association rules that you can view by using the context menu and selecting Browse.

Figure 2. Association rules

11 fields			
Model Settings Summary Annotations			
Consequent	Antecedent	Support %	Confidence %
frozenmeal	beer cannedveg	16.7	87.425
cannedveg	beer frozenmeal	17.0	85.882
beer	frozenmeal cannedveg	17.3	84.393

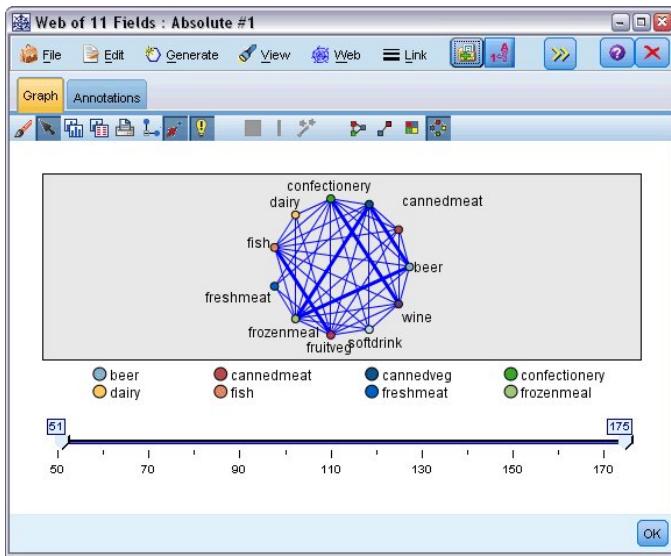
These rules show a variety of associations between frozen meals, canned vegetables, and beer. The presence of two-way association rules, such as:

```
frozenmeal -> beer
beer -> frozenmeal
```

suggests that a web display (which shows only two-way associations) might highlight some of the patterns in this data.

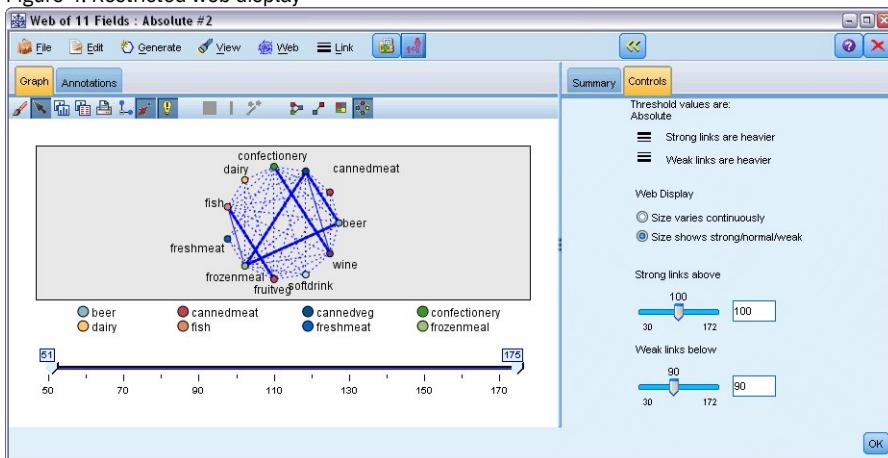
Attach a Web node to the Type node, edit the Web node, select all of the basket contents fields, select Show true flags only, and click run on the Web node.

Figure 3. Web display of product associations



Because most combinations of product categories occur in several baskets, the strong links on this web are too numerous to show the groups of customers suggested by the model.

Figure 4. Restricted web display



1. To specify weak and strong connections, click the yellow double arrow button on the toolbar. This expands the dialog box showing the web output summary and controls.
2. Select Size shows strong/normal/weak.
3. Set weak links below 90.
4. Set strong links above 100.

In the resulting display, three groups of customers stand out:

- Those who buy fish and fruits and vegetables, who might be called "healthy eaters"
- Those who buy wine and confectionery
- Those who buy beer, frozen meals, and canned vegetables ("beer, beans, and pizza")

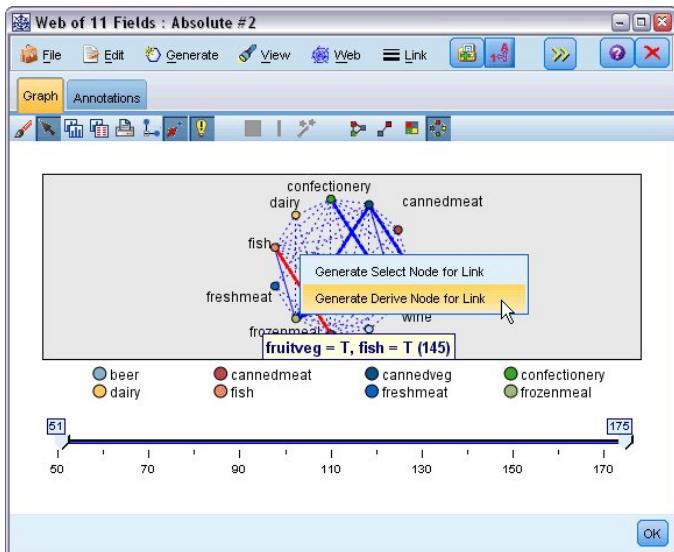
[Next](#)

Profiling the Customer Groups

You have now identified three groups of customers based on the types of products they buy, but you would also like to know who these customers are--that is, their demographic profile. This can be achieved by tagging each customer with a flag for each of these groups and using rule induction (C5.0) to build rule-based profiles of these flags.

First, you must derive a flag for each group. This can be automatically generated using the web display that you just created. Using the right mouse button, click the link between *fruitveg* and *fish* to highlight it, then right-click and select Generate Derive Node For Link.

Figure 1. Deriving a flag for each customer group



Edit the resulting Derive node to change the Derive field name to *healthy*. Repeat the exercise with the link from *wine* to *confectionery*, naming the resultant Derive field *wine_chocs*.

For the third group (involving three links), first make sure that no links are selected. Then select all three links in the *cannedveg*, *beer*, and *frozenmeal* triangle by holding down the shift key while you click the left mouse button. (Be sure you are in Interactive mode rather than Edit mode.) Then from the web display menus choose:

Generate...Derive Node ("And")

Change the name of the resultant Derive field to *beer_beans_pizza*.

To profile these customer groups, connect the existing Type node to these three Derive nodes in series, and then attach another Type node. In the new Type node, set the role of all fields to *None*, except for *value*, *pmethod*, *sex*, *hometown*, *income*, and *age*, which should be set to *Input*, and the relevant customer group (for example, *beer_beans_pizza*), which should be set to *Target*. Attach a C5.0 node, set the Output type to Rule set, and click run on the node. The resultant model (for *beer_beans_pizza*) contains a clear demographic profile for this customer group:

```
Rule 1 for T:
if sex = M
and income <= 16,900
then T
```

The same method can be applied to the other customer group flags by selecting them as the output in the second Type node. A wider range of alternative profiles can be generated by using Apriori instead of C5.0 in this context; Apriori can also be used to profile all of the customer group flags simultaneously because it is not restricted to a single output field.

[Next](#)

Summary

This example reveals how IBM® SPSS® Modeler can be used to discover affinities, or links, in a database, both by modeling (using Apriori) and by visualization (using a web display). These links correspond to groupings of cases in the data, and these groups can be investigated in detail and profiled by modeling (using C5.0 rule sets).

In the retail domain, such customer groupings might, for example, be used to target special offers to improve the response rates to direct mailings or to customize the range of products stocked by a branch to match the demands of its demographic base.

Assessing New Vehicle Offerings (KNN)

Nearest Neighbor Analysis is a method for classifying cases based on their similarity to other cases. In machine learning, it was developed as a way to recognize patterns of data without requiring an exact match to any stored patterns, or cases. Similar cases are near each other and dissimilar cases are distant from each other. Thus, the distance between two cases is a measure of their dissimilarity.

Cases that are near each other are said to be “neighbors.” When a new case (holdout) is presented, its distance from each of the cases in the model is computed. The classifications of the most similar cases – the nearest neighbors – are tallied and the new case is placed into the category that contains the greatest number of nearest neighbors.

You can specify the number of nearest neighbors to examine; this value is called k . The pictures show how a new case would be classified using two different values of k . When $k = 5$, the new case is placed in category 1 because a majority of the nearest neighbors belong to category 1. However, when $k = 9$, the new case is placed in category 0 because a majority of the nearest neighbors belong to category 0.

Nearest neighbor analysis can also be used to compute values for a continuous target. In this situation, the average or median target value of the nearest neighbors is used to obtain the predicted value for the new case.

An automobile manufacturer has developed prototypes for two new vehicles, a car and a truck. Before introducing the new models into its range, the manufacturer wants to determine which existing vehicles on the market are most like the prototypes--that is, which vehicles are their "nearest neighbors", and therefore which models they will be competing against.

The manufacturer has collected data about the existing models under a number of categories, and has added the details of its prototypes. The categories under which the models are to be compared include price in thousands (*price*), engine size (*engine_s*), horsepower (*horsepow*), wheelbase (*wheelbas*), width (*width*), length (*length*), curb weight (*curb_wgt*), fuel capacity (*fuel_cap*) and fuel efficiency (*mpg*).

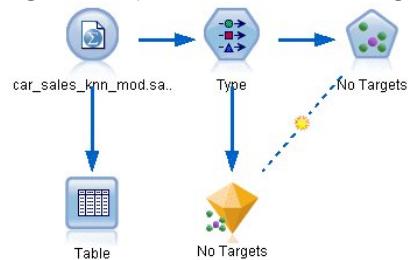
This example uses the stream named *car_sales_knn.str*, available in the *Demos* folder under the *streams* subfolder. The data file is *car_sales_knn_mod.sav*. See the topic [Demos Folder](#) for more information.

[Next](#)

- [Creating the Stream](#)
- [Examining the output](#)
- [Summary](#)

Creating the Stream

Figure 1. Sample stream for KNN modeling



Create a new stream and add a Statistics File source node pointing to *car_sales_knn_mod.sav* in the *Demos* folder of your IBM® SPSS® Modeler installation.

First, let's see what data the manufacturer has collected.

1. Attach a Table node to the Statistics File source node.
2. Open the Table node and click Run.

Figure 2. Source data for cars and trucks

	manufact	model	sales	resale	type	price	engine_s	horsepow	wheelbas	width
140	Toyota	Celica	33.269	15.445	0...	16...	1.800	140.000	102.400	68.3...
141	Toyota	Tacoma	64.087	9.575	1.0...	11...	2.400	142.000	103.300	66.5...
142	Toyota	Sienna	65.119	\$null\$	1.0...	22...	3.000	194.000	114.200	73.4...
143	Toyota	RAV4	25.106	13.325	1.0...	16...	2.000	127.000	94.900	66.7...
144	Toyota	4Run...	68.411	19.425	1.0...	22...	2.700	150.000	105.300	66.5...
145	Toyota	Land ...	9.835	34.080	1.0...	51...	4.700	230.000	112.200	76.4...
146	Volksw...	Golf	9.761	11.425	0.0...	14...	2.000	115.000	98.900	68.3...
147	Volksw...	Jetta	83.721	13.240	0.0...	16...	2.000	115.000	98.900	68.3...
148	Volksw...	Passat	51.102	16.725	0.0...	21...	1.800	150.000	106.400	68.5...
149	Volksw...	Cabrio	9.569	16.575	0.0...	19...	2.000	115.000	97.400	66.7...
150	Volksw...	GTI	5.596	13.760	0.0...	17...	2.000	115.000	98.900	68.3...
151	Volksw...	Beetle	49.463	\$null\$	0.0...	15...	2.000	115.000	98.900	67.9...
152	Volvo	S40	16.957	\$null\$	0.0...	23...	1.900	160.000	100.500	67.6...
153	Volvo	V40	3.545	\$null\$	0.0...	24...	1.900	160.000	100.500	67.6...
154	Volvo	S70	15.245	\$null\$	0.0...	27...	2.400	168.000	104.900	69.3...
155	Volvo	V70	17.531	\$null\$	0.0...	28...	2.400	168.000	104.900	69.3...
156	Volvo	C70	3.493	\$null\$	0.0...	45...	2.300	236.000	104.900	71.5...
157	Volvo	S80	18.969	\$null\$	0.0...	36...	2.900	201.000	109.900	72.1...
158	newC...	\$null\$	\$null\$	\$n...	21...	1.500	76.000	106.300	67.9...	
159	newT...	\$null\$	\$null\$	\$n...	34...	3.500	167.000	109.800	75.2...	

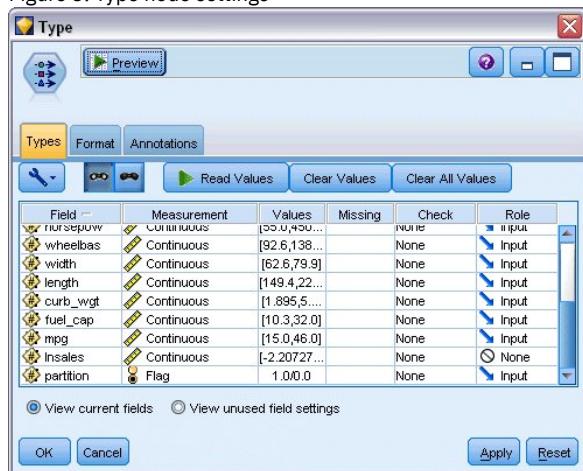
The details for the two prototypes, named *newCar* and *newTruck*, have been added at the end of the file.

We can see from the source data that the manufacturer is using the classification of "truck" (value of 1 in the *type* column) rather loosely to mean any non-automobile type of vehicle.

The last column, *partition*, is necessary in order that the two prototypes can be designated as holdouts when we come to identify their nearest neighbors. In this way, their data will not influence the calculations, as it is the rest of the market that we want to consider. Setting the *partition* value of the two holdout records to 1, while all the other records have a 0 in this field, enables us to use this field later when we come to set the focal records--the records for which we want to calculate the nearest neighbors.

Leave the table output window open for now, as we'll be referring to it later.

Figure 3. Type node settings

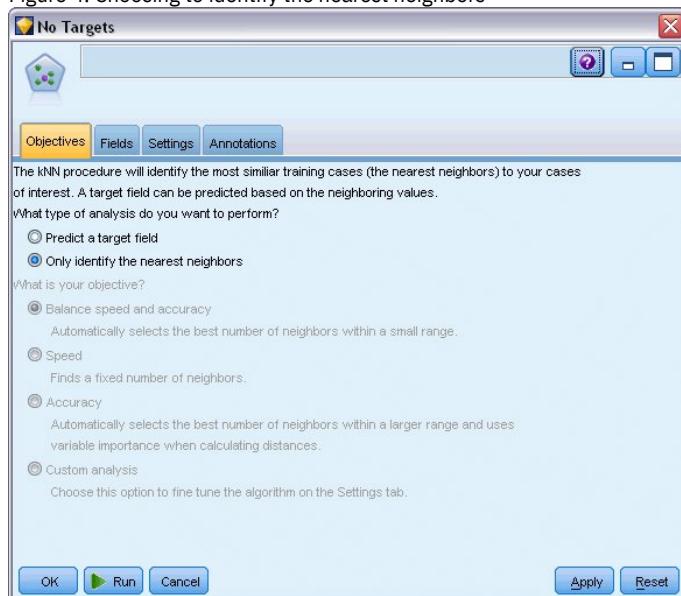


3. Add a Type node to the stream.
4. Attach the Type node to the Statistics File source node.
5. Open the Type node.

We want to make the comparison only on the fields *price* through *mpg*, so we'll leave the role for all these fields set to Input.

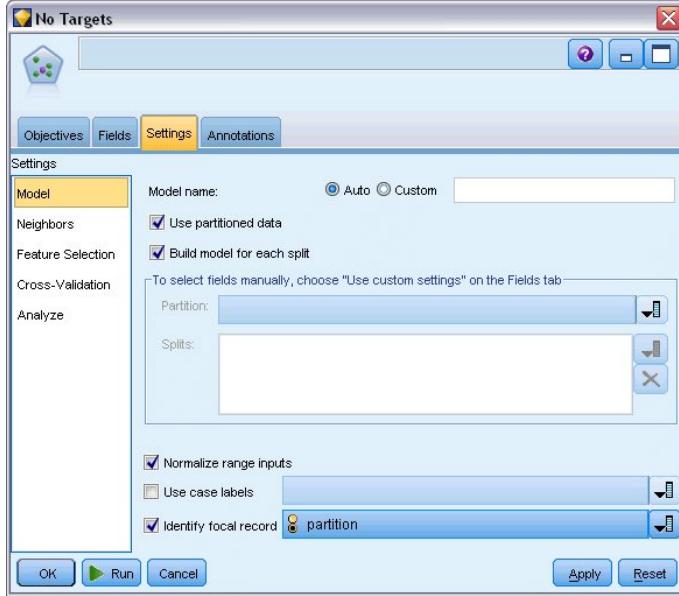
6. Set the role for all the other fields (*manufact* through *type*, plus *Insales*) to None.
7. Set the measurement level for the last field, *partition*, to Flag. Make sure that its role is set to Input.
8. Click Read Values to read the data values into the stream.
9. Click OK.

Figure 4. Choosing to identify the nearest neighbors



10. Attach a KNN node to the Type node.
 11. Open the KNN node.
- We're not going to be predicting a target field this time, because we just want to find the nearest neighbors for our two prototypes.
12. On the Objectives tab, choose Only identify the nearest neighbors.
 13. Click the Settings tab.

Figure 5. Using the partition field to identify the focal records



Now we can use the *partition* field to identify the focal records--the records for which we want to identify the nearest neighbors. By using a flag field, we ensure that records where the value of this field is set to 1 become our focal records.

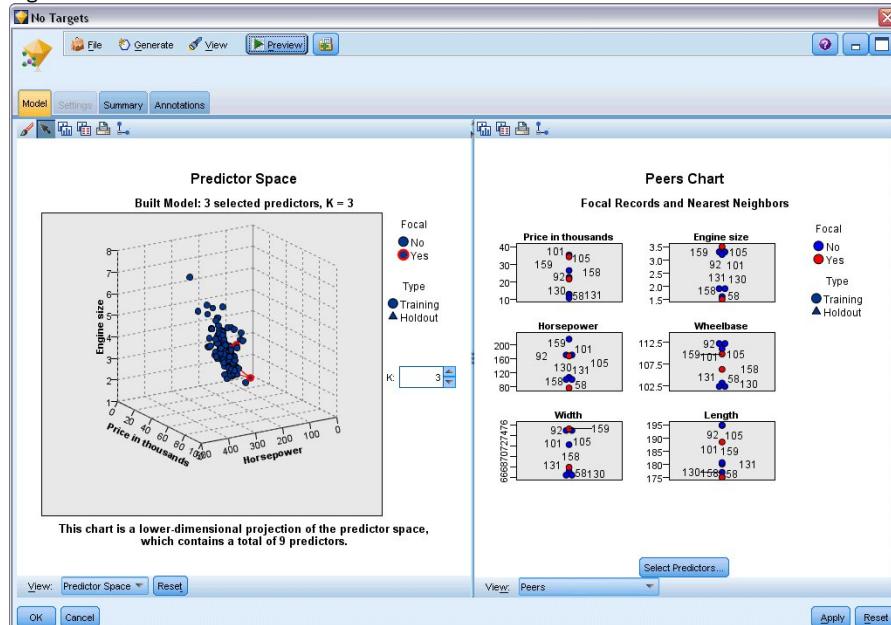
As we've seen, the only records that have a value of 1 in this field are *newCar* and *newTruck*, so these will be our focal records.

14. On the Model panel of the Settings tab, select the Identify focal record check box.
15. From the drop-down list for this field, choose partition.
16. Click the Run button.

[Next](#)

Examining the output

Figure 1. The Model Viewer window



A model nugget has been created on the stream canvas and in the Models palette. Open either of the nuggets to see the Model Viewer display, which has a two-panel window:

- The first panel displays an overview of the model called the main view. The main view for the Nearest Neighbor model is known as the **predictor space**.
- The second panel displays one of two types of views:

An auxiliary model view shows more information about the model, but is not focused on the model itself.

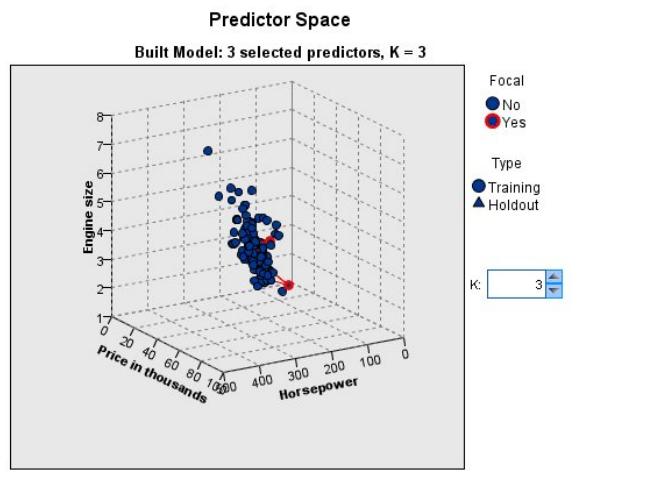
A linked view is a view that shows details about one feature of the model when you drill down on part of the main view.

[Next](#)

- [Predictor Space](#)
- [Peers Chart](#)
- [Neighbor and Distance Table](#)

Predictor Space

Figure 1. Predictor space chart



The predictor space chart is an interactive 3-D graph that plots data points for three features (actually the first three input fields of the source data), representing price, engine size and horsepower.

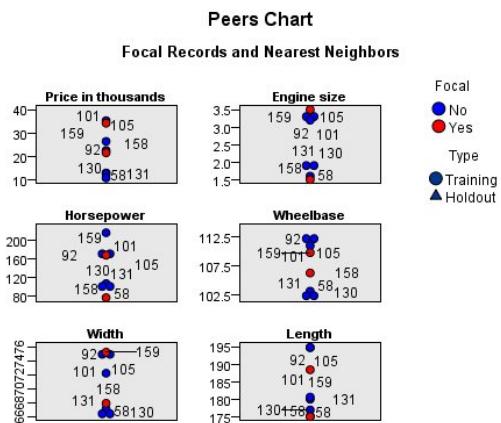
Our two focal records are highlighted in red, with lines connecting them to their k nearest neighbors.

By clicking and dragging the chart, you can rotate it to get a better view of the distribution of points in the predictor space. Click the Reset button to return it to the default view.

[Next](#)

Peers Chart

Figure 1. Peers chart



The default auxiliary view is the peers chart, which highlights the two focal records selected in the predictor space and their k nearest neighbors on each of six features--the first six input fields of the source data.

The vehicles are represented by their record numbers in the source data. This is where we need the output from the Table node to help identify them.

If the Table node output is still available:

1. Click the Outputs tab of the manager pane at the top right of the main IBM® SPSS® Modeler window.
 2. Double-click the entry Table (16 fields, 159 records).

If the table output is no longer available:

3. On the main IBM SPSS Modeler window, open the Table node.
4. Click Run.

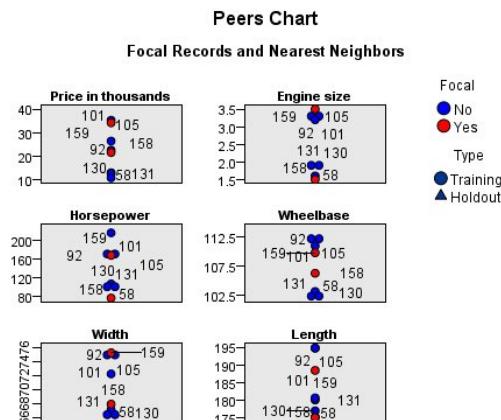
Figure 2. Identifying records by record number

Table (16 fields, 159 records)

	manufact	model	sales	resale	type	price	engine_s	horsepow	wheelbas	width
140	Toyota	Celica	33.269	15.445	0...0...	16.0...	1.800	140.000	102.400	68.3...
141	Toyota	Tacoma	84.087	9.575	1...0...	11.0...	2.400	142.000	103.300	66.5...
142	Toyota	Sienna	65.119	\$null\$	1...0...	22.0...	3.000	194.000	114.200	73.4...
143	Toyota	RAV4	25.106	13.325	1...0...	16.0...	2.000	127.000	94.900	66.7...
144	Toyota	4Run...	68.411	19.425	1...0...	22.0...	2.700	150.000	105.300	66.5...
145	Toyota	Land ...	9.835	34.080	1...0...	51.0...	4.700	230.000	112.200	76.4...
146	Volksw...	Golf	9.761	11.425	0...0...	14.0...	2.000	115.000	98.900	68.3...
147	Volksw...	Jetta	83.721	13.240	0...0...	16.0...	2.000	115.000	98.900	68.3...
148	Volksw...	Passat	51.102	16.725	0...0...	21.0...	1.800	150.000	106.400	68.5...
149	Volksw...	Cabrio	9.569	16.575	0...0...	19.0...	2.000	115.000	97.400	66.7...
150	Volksw...	GTI	5.596	13.760	0...0...	17.0...	2.000	115.000	98.900	68.3...
151	Volksw...	Beetle	49.463	\$null\$	0...0...	15.0...	2.000	115.000	98.900	67.9...
152	Volvo	S40	16.957	\$null\$	0...0...	23.0...	1.900	160.000	100.500	67.6...
153	Volvo	V40	3.545	\$null\$	0...0...	24.0...	1.900	160.000	100.500	67.6...
154	Volvo	S70	15.245	\$null\$	0...0...	27.0...	2.400	168.000	104.900	69.3...
155	Volvo	V70	17.531	\$null\$	0...0...	28.0...	2.400	168.000	104.900	69.3...
156	Volvo	C70	3.493	\$null\$	0...0...	45.0...	2.300	236.000	104.900	71.5...
157	Volvo	S80	18.969	\$null\$	0...0...	36.0...	2.900	201.000	109.900	72.1...
158		newC...	\$null\$	\$null\$	\$n...\$	21.0...	1.500	76.000	106.300	67.9...
159		newT...	\$null\$	\$null\$	\$n...\$	34.0...	3.500	167.000	109.800	75.2...

Scrolling down to the bottom of the table, we can see that *newCar* and *newTruck* are the last two records in the data, numbers 158 and 159 respectively.

Figure 3. Comparing features on the peers chart



From this we can see on the peers chart, for example, that *newTruck* (159) has a bigger engine size than any of its nearest neighbors, while *newCar* (158) has a smaller engine than any of *its* nearest neighbors.

For each of the six features, you can move the mouse over the individual dots to see the actual value of each feature for that particular case.

But which vehicles are the nearest neighbors for *newCar* and *newTruck*?

The peers chart is a little bit crowded, so let's change to a simpler view.

5. Click the View drop-down list at the bottom of the peers chart (the entry that currently says Peers).
 6. Select Neighbor and Distance Table.

[Next](#)

Neighbor and Distance Table

Figure 1. Neighbor and distance table

K Nearest Neighbors and Distances					
Focal Record	Displayed for Initial Focal Records				
	Nearest Neighbors			Nearest Distances	
	1	2	3	1	2
158	131	130	58	0.979	0.990
159	105	92	101	0.580	0.634

That's better. Now we can see the three models to which each of our two prototypes are closest in the market.

For *newCar* (focal record 158) they are the Saturn SC (131), the Saturn SL (130), and the Honda Civic (58).

No great surprises there--all three are medium-size saloon cars, so *newCar* should fit in well, particularly with its excellent fuel efficiency.

For *newTruck* (focal record 159), the nearest neighbors are the Nissan Quest (105), the Mercury Villager (92), and the Mercedes M-Class (101).

As we saw earlier, these are not necessarily trucks in the traditional sense, but simply vehicles that are classed as not being automobiles. Looking at the Table node output for its nearest neighbors, we can see that *newTruck* is relatively expensive, as well as being one of the heaviest of its type. However, fuel efficiency is again better than its closest rivals, so this should count in its favor.

[Next](#)

Summary

We've seen how you can use nearest-neighbor analysis to compare a wide-ranging set of features in cases from a particular data set. We've also calculated, for two very different holdout records, the cases that most closely resemble those holdouts.

Uncovering causal relationships in business metrics (TCM)

A business tracks various key performance indicators that describe the financial state of the business over time, and they also track a number of metrics that they can control. They are interested in using temporal causal modeling to uncover causal relationships between the controllable metrics and the key performance indicators. They would also like to know about any causal relationships among the key performance indicators.

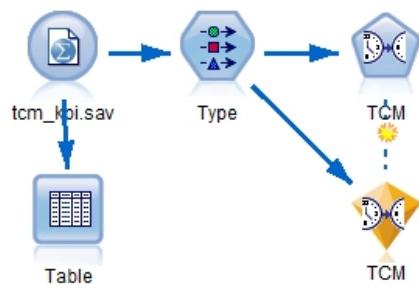
The data file tcm_kpi.sav contains weekly data on the key performance indicators and the controllable metrics. Data for the key performance indicators is stored in fields with the prefix *KPI*. Data for the controllable metrics is stored in fields with the prefix *Lever*.

[Next](#)

- [Creating the stream](#)
- [Running the analysis](#)
- [Overall Model Quality Chart](#)
- [Overall Model System](#)
- [Impact Diagrams](#)
- [Determining root causes of outliers](#)
- [Running scenarios](#)

Creating the stream

Figure 1. Sample stream for TCM modeling



1. Create a new stream and add a Statistics File source node pointing to *tcm_kpi.sav* in the *Demos* folder of your IBM® SPSS® Modeler installation.
2. Attach a Table node to the Statistics File source node.
3. Open the Table node and click Run to take a look at the data. It contains weekly data on the key performance indicators and the controllable metrics. Data for the key performance indicators is stored in fields with the prefix *KPI*, and data for the controllable metrics is stored in fields with the prefix *Lever*.

Figure 2. Source data for key performance indicators and controllable metrics

The screenshot shows a table with 31 fields and 112 records. The columns are labeled: date, Lever1, Lever2, Lever3, Lever4, Lever5, KPI_1, and KPI_2. The data is as follows:

	date	Lever1	Lever2	Lever3	Lever4	Lever5	KPI_1	KPI_2
1	2008-09-07	6.816	1.176	101.839	88.258	2027.711	1.829	1891.833
2	2008-09-14	6.091	1.172	120.610	103.803	2343.404	2.162	2125.261
3	2008-09-21	8.108	1.093	70.512	81.053	1813.224	1.809	1848.765
4	2008-09-28	6.503	1.121	78.581	86.393	2722.012	1.784	2551.153
5	2008-10-05	8.564	1.024	148.985	104.379	2235.634	1.704	2186.098
6	2008-10-12	7.331	0.848	170.236	91.477	2607.424	1.642	1711.295
7	2008-10-19	6.996	1.362	239.189	69.636	2354.322	1.681	2112.309
8	2008-10-26	7.863	0.959	169.925	87.400	1860.496	2.304	1561.226
9	2008-11-02	7.894	1.131	307.334	109.800	1600.156	1.782	1929.897
10	2008-11-09	6.548	1.052	467.642	77.574	2007.203	1.913	2042.415
11	2008-11-16	4.281	1.232	564.812	80.350	1764.707	1.915	2268.544
12	2008-11-23	7.458	1.219	523.018	105.373	2106.771	1.676	2451.158
13	2008-11-30	7.235	0.978	628.724	73.206	2666.294	2.160	2558.336
14	2008-12-07	7.752	1.032	654.648	99.905	1915.698	1.964	1614.402
15	2008-12-14	7.839	0.770	712.274	80.301	1811.261	1.147	1925.271
16	2008-12-21	8.529	1.374	699.621	98.391	1792.807	2.033	2320.790
17	2008-12-28	6.069	1.034	562.279	117.396	2216.657	0.879	2478.630
18	2009-01-04	6.174	1.442	613.071	72.062	2530.900	1.701	1769.694
19	2009-01-11	7.046	1.410	718.218	95.594	2285.149	1.841	2215.692
20	2009-01-18	5.805	0.933	908.362	83.863	2391.528	1.977	2094.555

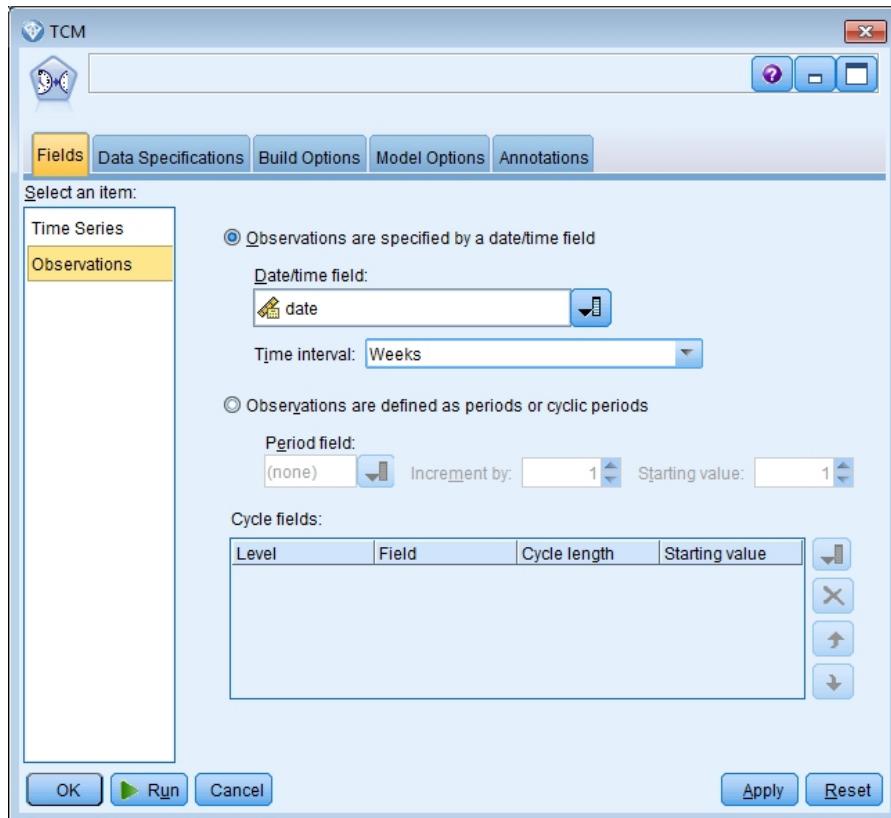
4. Add a Type node to the stream.
5. Attach the Type node to the Statistics File source node.

[Next](#)

Running the analysis

1. Attach a TCM node to the Type node, then open the TCM node and go to the Observations section of the Fields tab.

Figure 1. Temporal Causal Modeling, observations



2. Select *date* from the Date/time field and select *Weeks* from the Time interval field.

3. Click Time Series and select Use predefined roles.

In the sample data set *tcm_kpi.sav*, the fields *Lever1* through *Lever5* have the role of Input and *KPI_1* through *KPI_25* have the role of Both. When Use predefined roles is selected, fields with a role of Input are treated as candidate inputs and fields with a role of Both are treated as both candidate inputs and targets for temporal causal modeling.

The temporal causal modeling procedure determines the best inputs for each target from the set of candidate inputs. In this example, the candidate inputs are the fields *Lever1* through *Lever5* and the fields *KPI_1* through *KPI_25*.

4. Click Run.

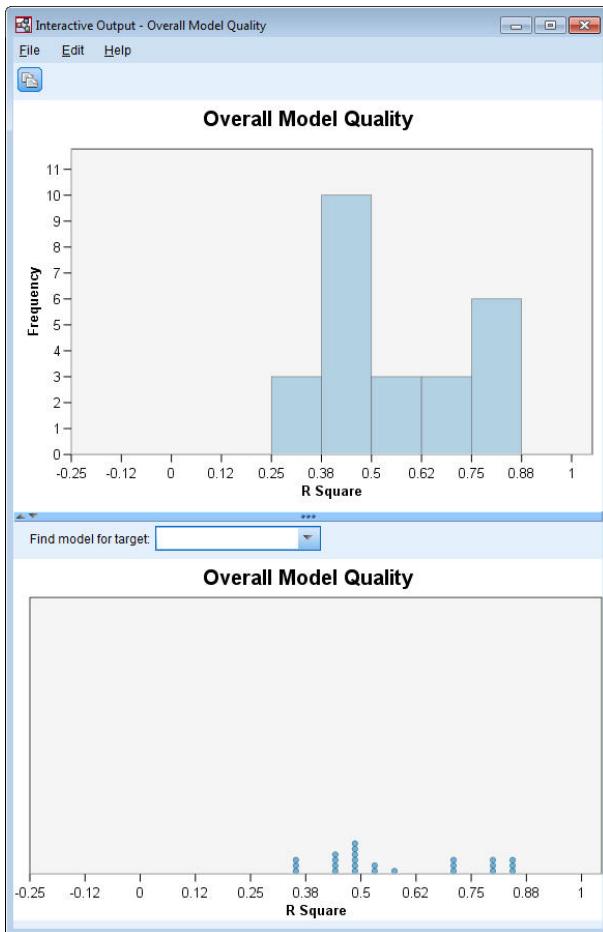
[Next](#)

Overall Model Quality Chart

The Overall Model Quality output item, which is generated by default, displays a bar chart and an associated dot plot of the model fit for all models. There is a separate model for each target series. The model fit is measured by the chosen fit statistic. This example uses the default fit statistic, which is R Square.

The Overall Model Quality item contains interactive features. To enable the features, activate the item by double-clicking the Overall Model Quality chart in the Viewer.

Figure 1. Overall Model Quality



Clicking a bar in the bar chart filters the dot plot so that it displays only the models that are associated with the selected bar. Hovering over a dot in the dot plot displays a tooltip that contains the name of the associated series and the value of the fit statistic. You can find the model for a particular target series in the dot plot by specifying the series name in the Find model for target box.

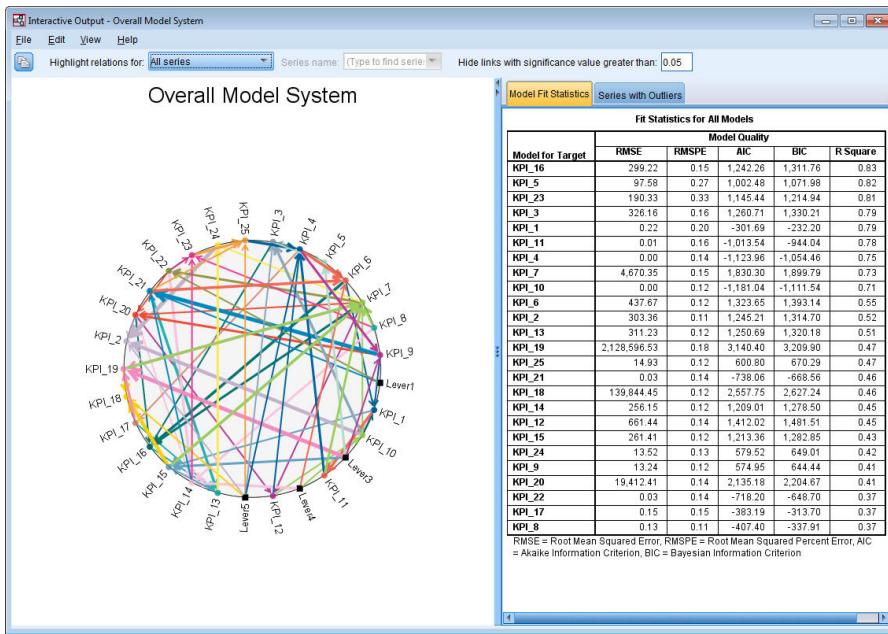
[Next](#)

Overall Model System

The Overall Model System output item, which is generated by default, displays a graphical representation of the causal relations between series in the model system. By default, relations for the top 10 models are shown, as determined by the value of the R Square fit statistic. The number of top models (also referred to as best-fitting models) and the fit statistic are specified on the Series to Display settings (on the Build Options tab) of the Temporal Causal Modeling dialog.

The Overall Model System item contains interactive features. To enable the features, activate the item by double-clicking the Overall Model System chart in the Viewer. In this example, it is most important to see the relations between all series in the system. In the interactive output, select All series from the Highlight relations for drop-down list.

Figure 1. Overall Model System, view for all series



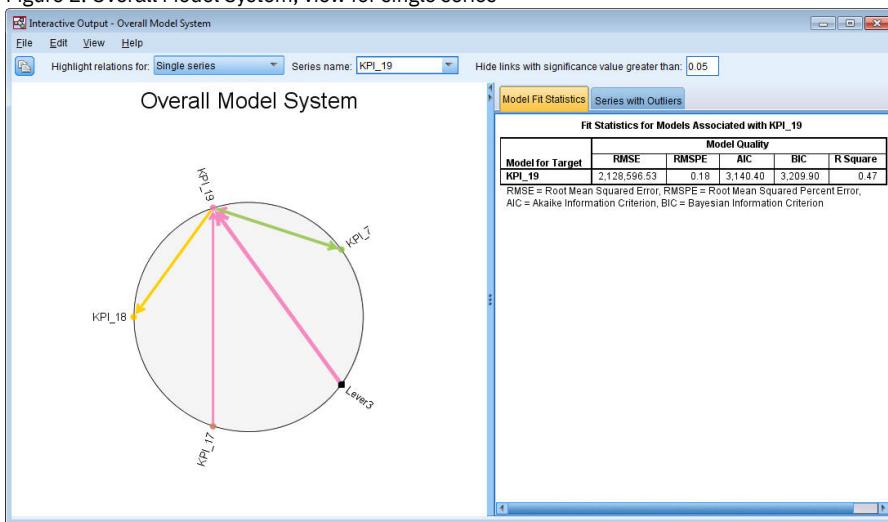
All lines that connect a particular target to its inputs have the same color, and the arrow on each line points from an input to the target of that input. For example, *Lever3* is an input to *KPI_19*.

The thickness of each line indicates the significance of the causal relation, where thicker lines represent a more significant relation. By default, causal relations with a significance value greater than 0.05 are hidden. At the 0.05 level, only *Lever1*, *Lever3*, *Lever4*, and *Lever5* have significant causal relations with the key performance indicator fields. You can change the threshold significance level by entering a value in the field that is labeled Hide links with significance value greater than.

In addition to uncovering causal relations between *Lever* fields and key performance indicator fields, the analysis also uncovered relations among the key performance indicator fields. For example, *KPI_10* was selected as an input to the model for *KPI_2*.

You can filter the view to show only the relations for a single series. For example, to view only the relations for *KPI_19*, click the label for *KPI_19*, right-click, and select Highlight relations for series.

Figure 2. Overall Model System, view for single series



This view shows the inputs to *KPI_19* that have a significance value less than or equal to 0.05. It also shows that, at the 0.05 significance level, *KPI_19* was selected as an input to both *KPI_18* and *KPI_7*.

In addition to displaying the relations for the selected series, the output item also contains information about any outliers that were detected for the series. Click the Series with Outliers tab.

Figure 3. Outliers for KPI_19
Series with Outliers for KPI_19

Series	Time	Observed Value
KPI_19	2008-10-12	7,358,201.68
	2009-04-05	2.10E+007
	2010-09-19	6,492,157.97

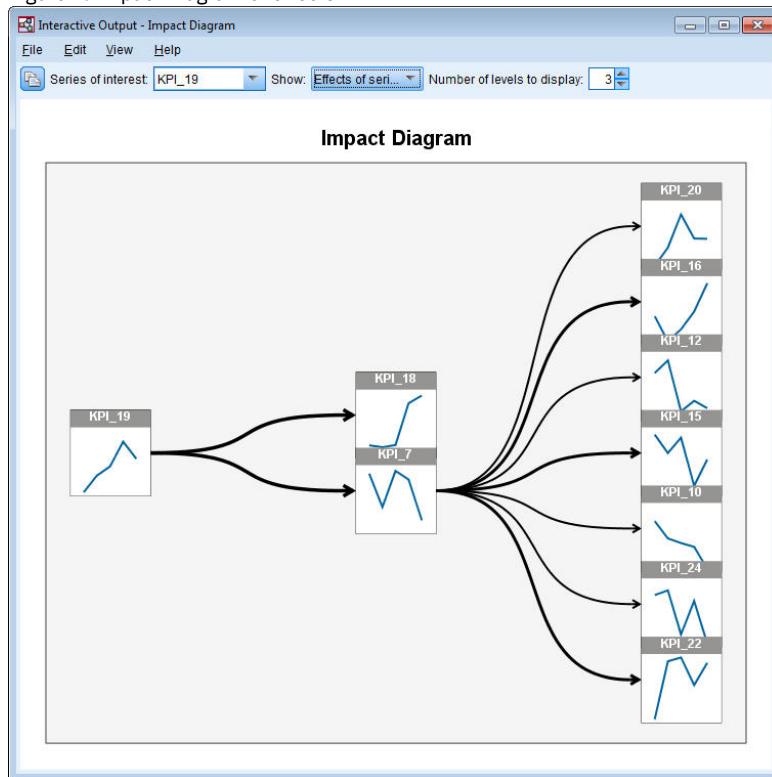
Three outliers were detected for *KPI_19*. Given the model system, which contains all of the discovered connections, it is possible to go beyond outlier detection and determine the series that most likely causes a particular outlier. This type of analysis is referred to as outlier root cause analysis and is covered in a later topic in this case study.

[Next](#)

Impact Diagrams

You can obtain a complete view of all relations that are associated with a particular series by generating an impact diagram. Click the label for *KPI_19* in the Overall Model System chart, right-click, and select Create Impact Diagram.

Figure 1. Impact Diagram of effects



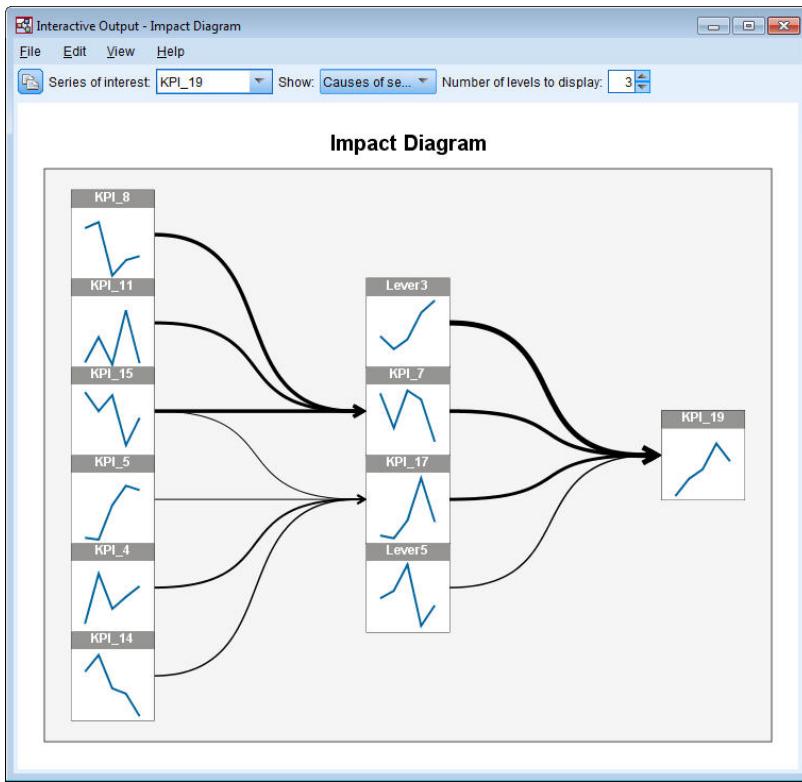
When an impact diagram is created from the Overall Model System, as in this example, it initially shows the series that are affected by the selected series. By default, impact diagrams show three levels of effects, where the first level is just the series of interest. Each additional level shows more indirect effects of the series of interest. You can change the value of the Number of levels to display to show more or fewer levels of effects. The impact diagram for this example shows that *KPI_19* is a direct input to both *KPI_18* and *KPI_7*, but it indirectly affects a number of series through its effect on series *KPI_7*. As in the overall model system, the thickness of the lines indicates the significance of the causal relations.

The chart that is displayed in each node of the impact diagram shows the last $L+1$ values of the associated series at the end of the estimation period and any forecast values, where L is the number of lag terms that are included in each model. You can obtain a detailed sequence chart of these values by single-clicking the associated node.

Double-clicking a node sets the associated series as the series of interest, and regenerates the impact diagram based on that series. You can also specify a series name in the Series of interest box to select a different series of interest.

Impact diagrams can also show the series that affect the series of interest. These series are referred to as *causes*. To see the series that affect *KPI_19*, select Causes of series from the Show drop-down.

Figure 2. Impact Diagram of causes



This view shows that the model for *KPI_19* has four inputs and that *Lever3* has the most significant causal connection with *KPI_19*. It also shows series that indirectly affect *KPI_19* through their effects on *KPI_7* and *KPI_17*. The same concept of levels that was discussed for effects also applies to causes. Likewise, you can change the value of the Number of levels to display to show more or fewer levels of causes.

[Next](#)

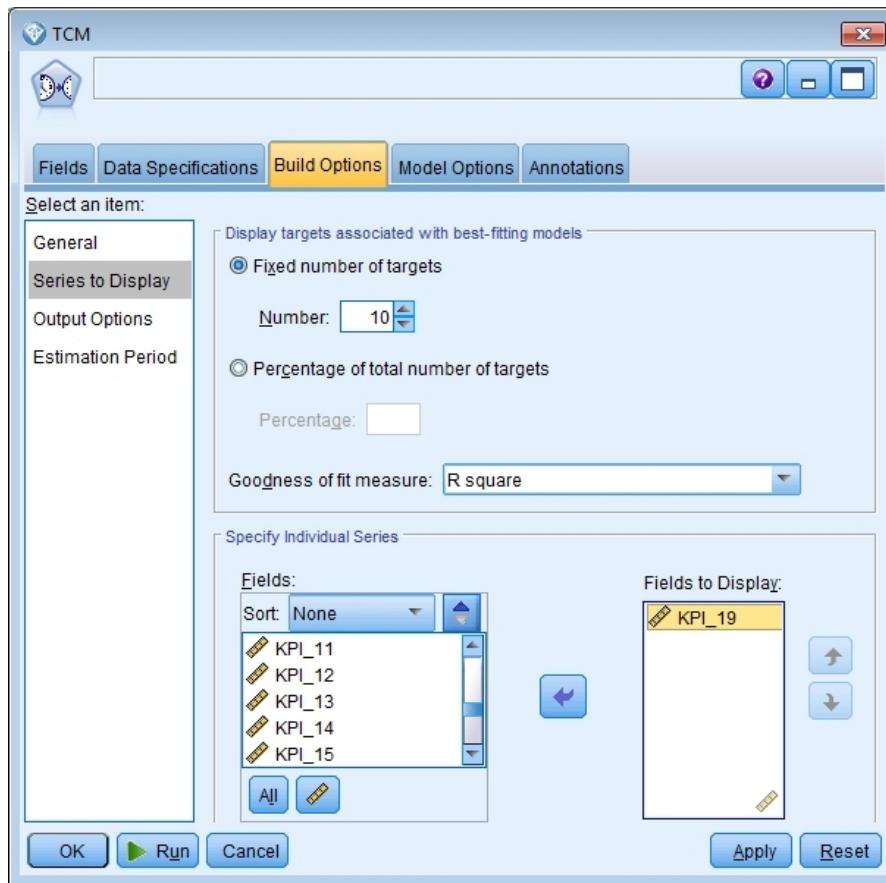
Determining root causes of outliers

Given a temporal causal model system, it is possible to go beyond outlier detection and determine the series that most likely causes a particular outlier. This process is referred to as outlier root cause analysis and must be requested on a series by series basis. The analysis requires a temporal causal model system and the data that was used to build the system. In this example, the active dataset is the data that was used to build the model system.

To run outlier root cause analysis:

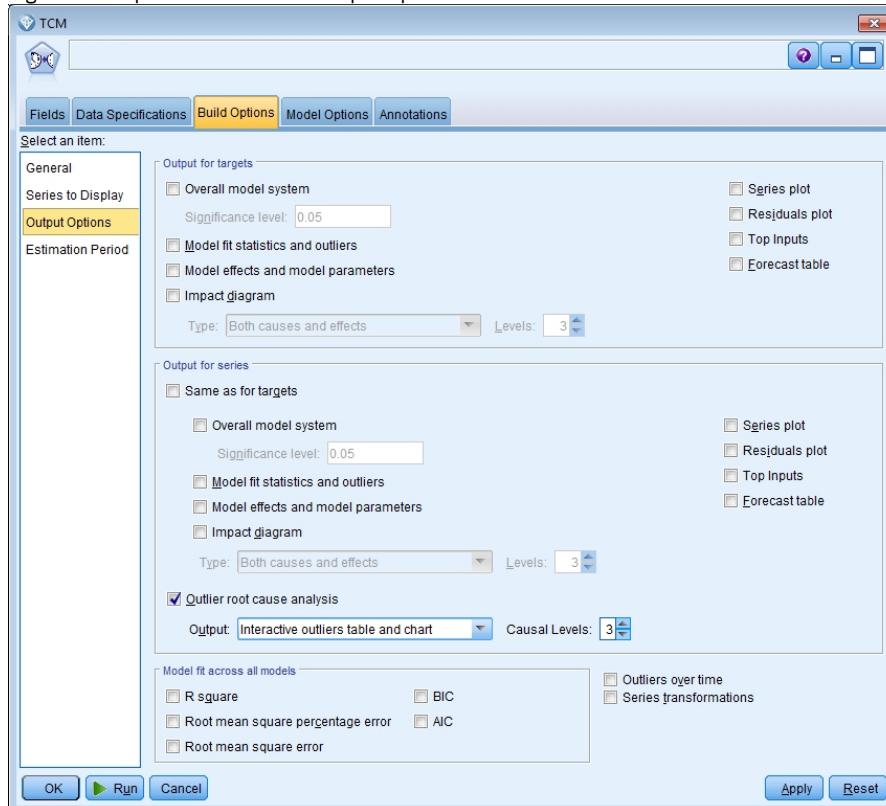
1. In the TCM dialog, go to the Build Options tab and then click Series to Display in the Select an item list.

Figure 1. Temporal Causal Model Series to Display



2. Move *KPI_19* to the Fields to display list.
3. Click Output options in the Select an item list on the Options tab.

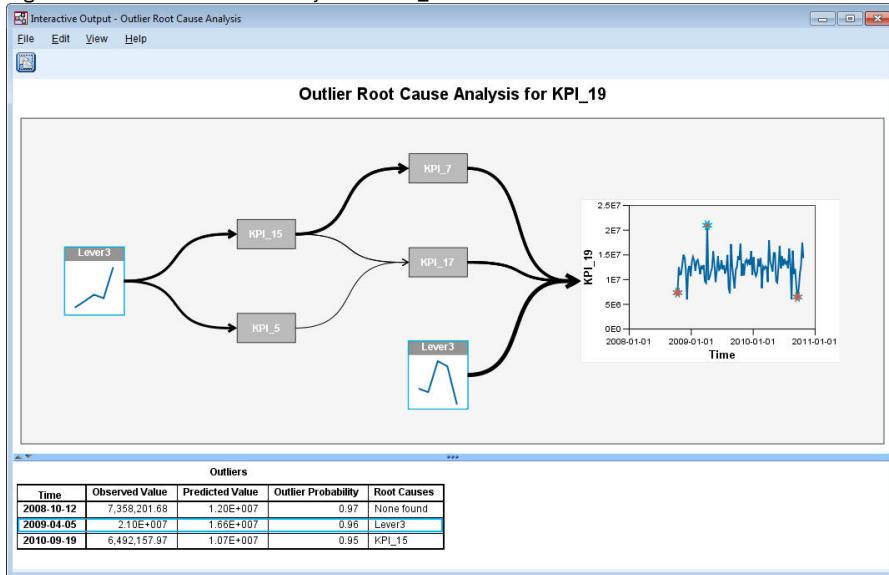
Figure 2. Temporal Causal Model Output Options



4. Deselect Overall model system, Same as for targets, R square, and Series transformations.
5. Select Outlier root cause analysis and keep the existing settings for Output and Causal levels.
6. Click Run.

7. Double-click the Outlier Root Cause Analysis chart for *KPI_19* in the Viewer to activate it.

Figure 3. Outlier Root Cause Analysis for *KPI_19*



The results of the analysis are summarized in the Outliers table. The table shows that root causes were found for the outliers at 2009-04-05 and 2010-09-19, but no root cause was found for the outlier at 2008-10-12. Clicking a row in the Outliers table highlights the path to the root cause series, as shown here for the outlier at 2009-04-05. This action also highlights the selected outlier in the sequence chart. You can also click the icon for an outlier directly in the sequence chart to highlight the path to the root cause series for that outlier.

For the outlier at 2009-04-05, the root cause is *Lever3*. The diagram shows that *Lever3* is a direct input to *KPI_19*, but that it also indirectly influences *KPI_19* through its effect on other series that affect *KPI_19*. One of the configurable parameters for outlier root cause analysis is the number of causal levels to search for root causes. By default, three levels are searched. Occurrences of the root cause series are displayed up to the specified number of causal levels. In this example, *Lever3* occurs at both the first causal level and the third causal level.

Each node in the highlighted path for an outlier contains a chart whose time range depends on the level at which the node occurs. For nodes in the first causal level, the range is T-1 to T-L where T is the time at which the outlier occurs and L is the number of lag terms that are included in each model. For nodes in the second causal level, the range is T-2 to T-L-1; and for the third level the range is T-3 to T-L-2. You can obtain a detailed sequence chart of these values by single-clicking the associated node.

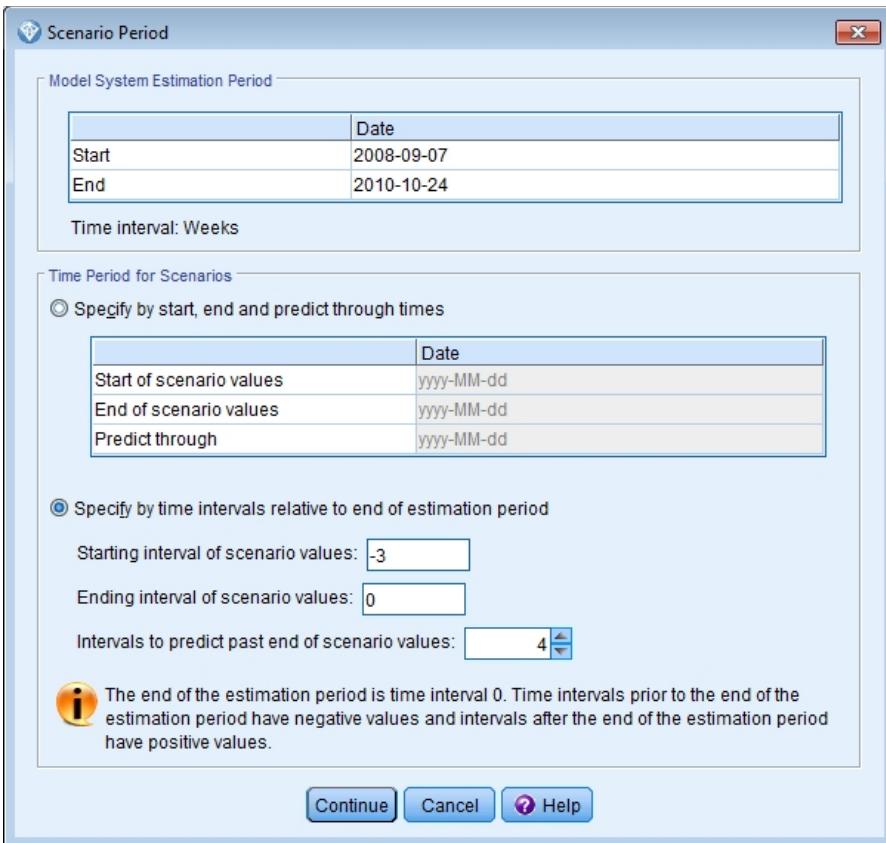
Running scenarios

Given a temporal causal model system, you can run user-defined scenarios. A *scenario* is defined by a time series, that is referred to as the *root series*, and a set of user-defined values for that series over a specified time range. The specified values are then used to generate predictions for the time series that are affected by the root series. The analysis requires a temporal causal model system and the data that was used to build the system. In this example, the active dataset is the data that was used to build the model system.

To run scenarios:

1. In the TCM output dialog, click the Scenario Analysis button.
2. In the Temporal Causal Model Scenarios dialog, click Define Scenario Period.

Figure 1. Scenario Period



3. Select Specify by time intervals relative to end of estimation period.
4. Enter **-3** for the starting interval and enter **0** for the ending interval.

These settings specify that each scenario is based on values that are specified for the last four time intervals in the estimation period. For this example, the last four time intervals means the last four weeks. The time range over which the scenario values are specified is referred to as the *scenario period*.

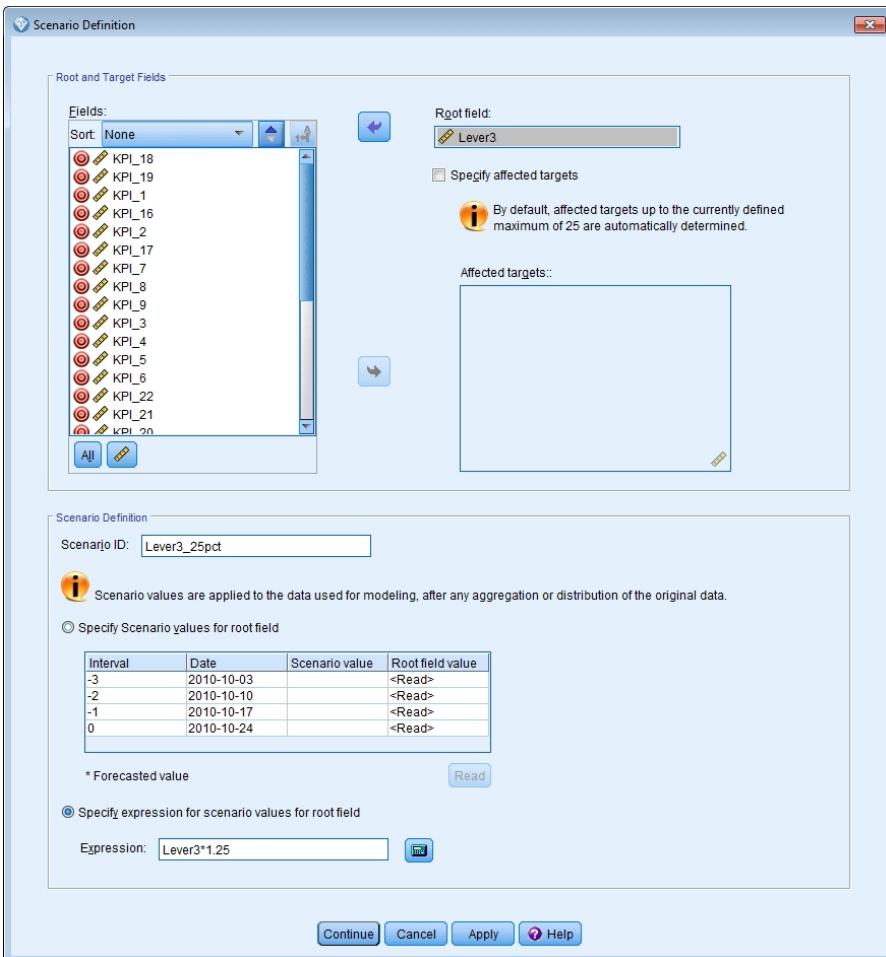
5. Enter **4** for the intervals to predict past the end of the scenarios values.

This setting specifies that predictions are generated for four time intervals beyond the end of the scenario period.

6. Click Continue.

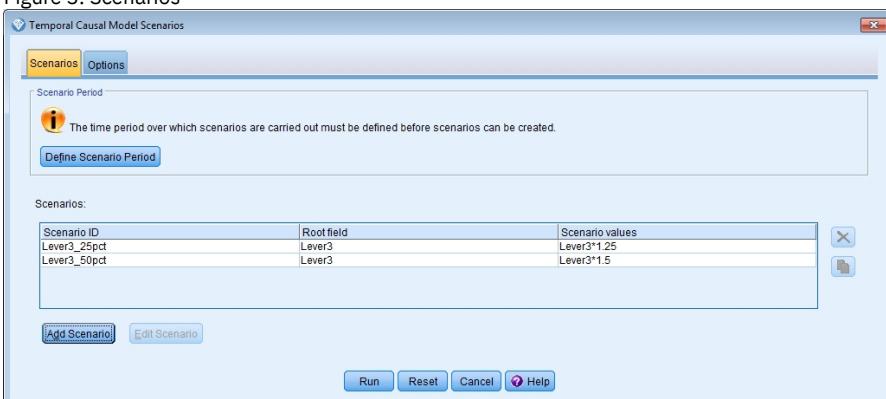
7. Click Add Scenario on the Scenarios tab.

Figure 2. Scenario Definition



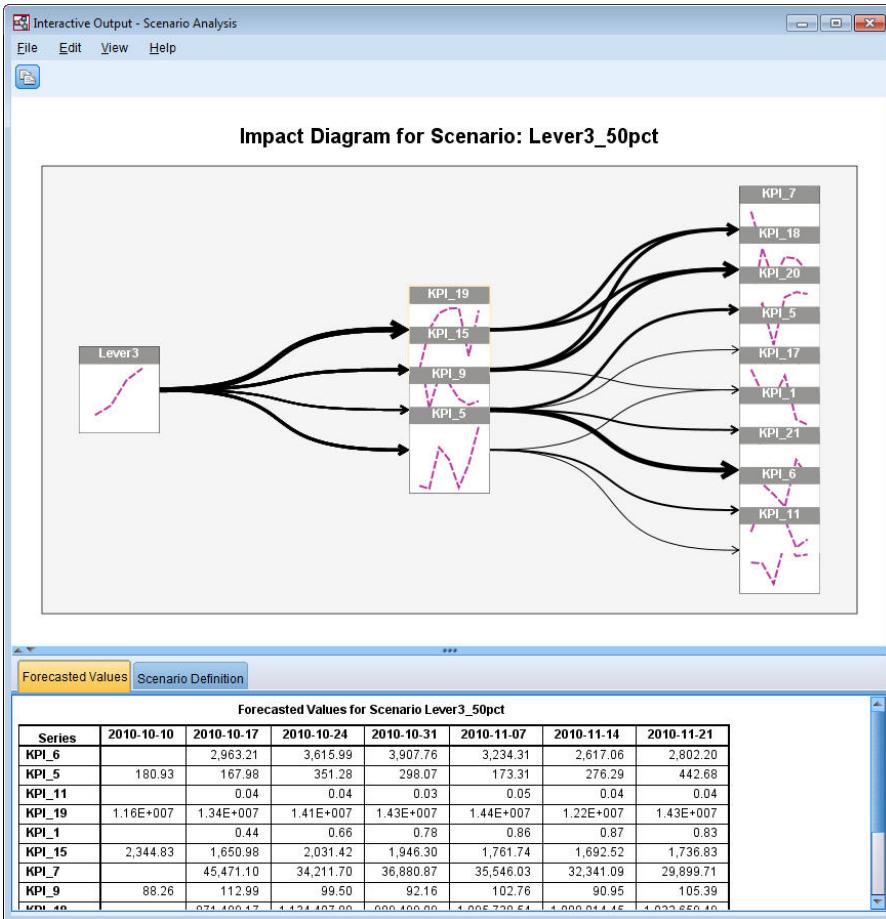
8. Move *Lever3* to the Root Field box to examine how specified values of *Lever3* in the scenario period affect predictions of the other series that are causally affected by *Lever3*.
9. Enter *Lever3_25pct* for the scenario ID.
10. Select Specify expression for scenario values for root field and enter *Lever3*1.25* for the expression.
This setting specifies that the values for *Lever3* in the scenario period are 25% larger than the observed values. For more complex expressions, you can use the Expression Builder by clicking the calculator icon.
11. Click Continue.
12. Repeat steps 10 - 14 to define a scenario that has *Lever3* for the root field, *Lever3_50pct* for the scenario ID, and *Lever3*1.5* for the expression.

Figure 3. Scenarios



13. Click the Options tab and enter 2 for the maximum level for affected targets.
14. Click Run.
15. Double-click the Impact Diagram chart for *Lever3_50pct* in the Viewer to activate it.

Figure 4. Impact Diagram for Scenario: *Lever3_50pct*

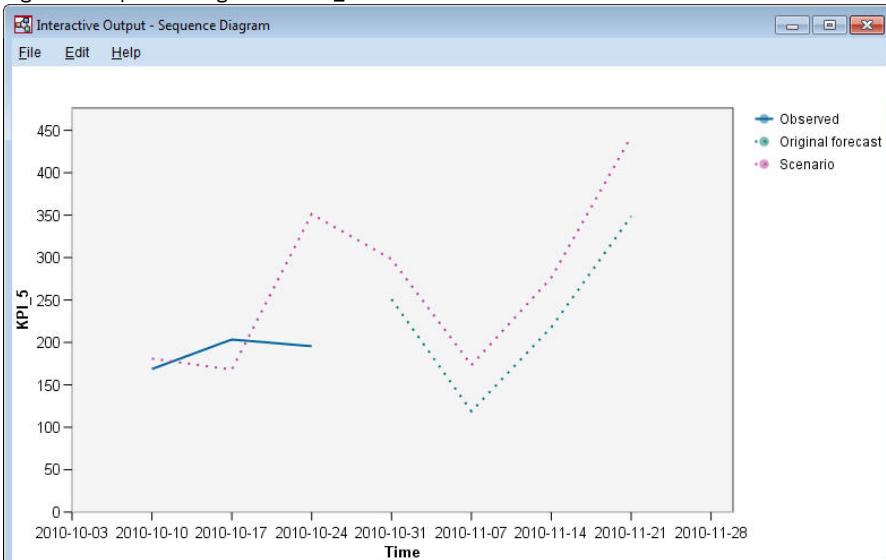


The Impact Diagram shows the series that are affected by the root series *Lever3*. Two levels of effects are shown because you specified 2 for the maximum level for affected targets.

The Forecasted Values table includes the predictions for all of the series that are affected by *Lever3*, up to the second level of effects. Predictions for target series in the first level of effects start at the first time period after the beginning of the scenario period. In this example, predictions for target series in the first level start at 2010-10-10. Predictions for target series in the second level of effects start at the second time period after the beginning of the scenario period. In this example, predictions for target series in the second level start at 2010-10-17. The staggered nature of the predictions reflects the fact that the time series models are based on lagged values of the inputs.

- Click the node for *KPI_5* to generate a detailed sequence diagram.

Figure 5. Sequence Diagram for KPI_5



The sequence chart shows the predicted values from the scenario, and it also shows the values of the series in the absence of the scenario. When the scenario period contains times within the estimation period, the observed values of the series are shown. For times beyond the end of the estimation period, the original forecasts are shown.

Glossary

P

practical significance

A statistical test will answer the question, "Is there a difference between two groups?" but not the follow-up "Is that difference large enough for me to care?" It is up to you to determine whether test results are useful to your situation.

IBM SPSS Modeler Gold

- [Overview](#)
 - [Prerequisites](#)
 - [Installing IBM SPSS Modeler Gold](#)
 - [Post-installation steps](#)
 - [Passport Advantage part numbers](#)
-

Overview

This guide includes information about installing and configuring the SPSS product components for IBM® SPSS® Modeler Gold version 18.4.

SPSS Modeler Gold is a suite of the following SPSS products:

- IBM SPSS Collaboration and Deployment Services version 8.4
- IBM SPSS Modeler version 18.4

This guide contains general steps for installing and configuring these products and their various adapters and components, with links to existing installation instructions.

Documentation for all IBM SPSS products is available on the [IBM Documentation](#) site. Search for SPSS. Documentation is also available in PDF format at the following locations:

- [IBM SPSS Collaboration and Deployment Services 8.4 PDF documentation](#)
- [IBM SPSS Modeler 18.4 PDF documentation](#)

SPSS Modeler Gold topology

The following boxes represent the machines that make up a recommended SPSS Modeler Gold deployment, and list all the possible components that can be installed. Depending on your environment and your needs, note that your specific deployment may be organized differently, and may not include every possible software component. This is one example, and these machines will be referred to as *Server 1*, *Server 2*, and *Clients* throughout this document.

Figure 1. SPSS Modeler Gold Topology

Server 1 (IBM SPSS Collaboration and Deployment Services machine)

- IBM Installation Manager
- A supported web application server (WebSphere, for example)
- A supported database (IBM DB2, for example)
- IBM SPSS Collaboration and Deployment Services Repository Server
- IBM SPSS Collaboration and Deployment Services Scoring Adapter for PMML
- IBM SPSS Modeler Adapter for IBM SPSS Collaboration and Deployment Services

Server 2 (IBM SPSS Modeler Server machine)

- IBM SPSS Modeler Server
- IBM SPSS Modeler Essentials for R
- IBM SPSS Modeler Text Analytics Server
- IBM SPSS Data Access Pack

Clients

- IBM SPSS Deployment Manager client (includes the Administration Consoles)
- IBM SPSS Modeler client
- IBM SPSS Modeler Essentials for R
- IBM SPSS Modeler Scoring Adapter
- IBM SPSS Modeler Text Analytics Client
- IBM SPSS Collaboration and Deployment Services – Essentials for Python

Prerequisites

Before installing and configuring the IBM® SPSS® Modeler Gold products, it is assumed the environment meets all prerequisite system requirements such as a supported application server and database. Review the system requirements for each IBM SPSS product you plan to install before proceeding (for example, to ensure your application server and database are supported by all IBM SPSS components to be installed).

The following prerequisites are required:

IBM Installation Manager

IBM Installation Manager version 1.9.1 is required on *Server 1* for installing many of the products in the suite. For more information, see the [Installation Manager documentation](#).

Web Application Server

You must install and configure a supported application server. For example, if using WebSphere, configure the following items:

- Create a WebSphere profile (called *AppSrv1*, for example)
- Keep the default security active
- Create a WebSphere admin user (called *wsadmin*, for example)

Database

You must install and configure your database to be used as a data source, to host the IBM SPSS Collaboration and Deployment Services Repository. A repository database must be running and accessible before you install and configure the IBM SPSS Collaboration and Deployment Services Repository server.

For example, if using IBM Db2, configure the following items:

- Create a Db2 admin user (called *db2admin*, for example)
- Create database users if they'll be required later
- Create an IBM SPSS Collaboration and Deployment Services Repository database instance by running the example Db2 script included with IBM SPSS Collaboration and Deployment Services. See [this documentation](#) for details.

Note: For complete instructions and details about database requirements, see the [the IBM SPSS Collaboration and Deployment Services documentation](#).

Installation files

Installation files for all SPSS Modeler Gold products to be installed must be downloaded from Passport Advantage and copied to the appropriate machine. Or you can point to the Passport Advantage site during installation for some components. For a list of the required Passport Advantage part numbers, and which machine to place each downloaded file on, see [Passport Advantage part numbers](#).

Microsoft Silverlight

If you use Microsoft Internet Explorer to view visualizations, you must have Microsoft Silverlight 5 or later installed on your computer. Firefox or iPad users do not need Silverlight.

Installing IBM® SPSS® Modeler Gold

The installation instructions here are organized by the machines the products will be installed on. See the [Overview](#) for a recommendation of which components to install on which machine. We recommend following the installation order outlined in this section.

- [IBM SPSS Collaboration and Deployment Services Server components \(Server 1\)](#)
 - [IBM SPSS Modeler Server components \(Server 2\)](#)
 - [Client components](#)
 - [Optional components](#)
-

IBM SPSS Collaboration and Deployment Services Server components (Server 1)

Complete the following steps to install components on *Server 1*.

Before you can install, IBM Installation Manager must have access to the repository that contains the IBM SPSS product packages. You must also shut down your application server.

If you are installing from a repository that is not on the Passport Advantage® site, you must specify the repository in the preferences before you install. For more information, see [Repository preferences](#).

1. Start Installation Manager in wizard mode by running the `IBMMIM` application file. For more information, see [Start Installation Manager](#).
2. In Installation Manager, click Install. Installation Manager searches the defined repositories for available packages. If no available packages are found, verify that you specified repository preferences correctly.
3. The Install page of Installation Manager lists all the packages that were found in the repository that Installation Manager searched. Installation Manager will check Fix Central on the Web for the latest fix pack versions of the packages. Be sure you install the latest fix packs for all SPSS components installed.
4. Select the following packages and then click Next.
 - IBM SPSS Collaboration and Deployment Services - Repository Server (base offering)
 - IBM SPSS Collaboration and Deployment Services Scoring Adapter for PMML
 - IBM® SPSS® Modeler Adapter for IBM SPSS Collaboration and Deployment Services

Note: IBM SPSS Collaboration and Deployment Services - Repository Server is the base offering. The other packages are extensions and will be installed on top of (into the directory of) the base Repository Server offering.

5. On the Licenses page, accept the license agreement and click Next to continue.
 6. On the Location page, enter the path for the shared resources directory in the Shared Resources Directory field. The shared resources directory contains resources that can be shared by multiple package groups. Click Next.
 7. Click Next to continue the installation.
 8. On the next Location page, select the translations to install for packages in the package group. The corresponding language translations for the graphical user interface and documentation are installed. Choices apply to all packages that are installed in this package group. This option may not apply to all product installations. Click Next to continue.
 9. The Features page shows the package features that will be installed. Accept all defaults. You can click a feature to view its brief description under Details.
 10. On the Summary page, review your choices before you install the packages.
- On Windows, Installation Manager checks for running processes. If processes are blocking the installation, a list of these processes is shown in the Blocking Processes section. You must stop these processes before you continue the installation. Click Stop All Blocking Processes. If there are no processes that must be stopped, you do not see this list. The running processes lock files that must be accessed or modified by Installation Manager.

11. Click Install. When the installation process completes, you receive a confirmation message.
 12. Run the Repository configuration utility. See the instructions online [here](#).
 13. Start the WebSphere application server and then use the Windows Control Panel to start the IBM SPSS Collaboration and Deployment Services Server. Go to Control Panel, then Administrative Tools, then Services.
 14. Wait a few minutes and then verify the Repository server status by launching browser-based IBM SPSS Collaboration and Deployment Services Deployment Manager. See the instructions [here](#).
-

IBM SPSS Modeler Server components (Server 2)

Complete the following steps to install components on *Server 2*.

Important: Install the latest Fix Packs for all components.

1. Install SPSS® Modeler Server. See the installation instructions available online [here](#) for Windows or [here](#) for UNIX.
 2. Edit the SPSS Modeler Server options.cfg file, if necessary (for example, to update the server port number). See the documentation available online [here](#).
 3. Install IBM® SPSS Modeler Text Analytics Server into SPSS Modeler Server. See the SPSS Modeler Premium installation instructions available online [here](#).
 4. Install SPSS Modeler Essentials for R. See the installation instructions available online [here](#). Note in the instructions that you must first download and install R from [here](#).
 5. Install the IBM SPSS Data Access Pack. See the installation instructions available online [here](#) for Windows or [here](#) for UNIX.
 6. Use the Windows Control Panel to start the SPSS Modeler Server and the Text Analytics Server. Go to Control Panel, then Administrative Tools, then Services.
-

Client components

Complete the following steps to install client software.

Important: Install the latest Fix Packs for all components.

1. Install IBM® SPSS® Deployment Manager Client. See the installation instructions available online [here](#).
Note: Deployment Manager includes the administration consoles.
 2. Install IBM SPSS Modeler Client. See the installation instructions available online [here](#).
 3. Install SPSS Modeler Essentials for R. See the installation instructions available online [here](#). Note in the instructions that you must first install an R environment from [here](#).
 4. Install IBM SPSS Modeler Text Analytics. See the SPSS Modeler Premium Client installation instructions available online [here](#).
 5. If you want to use ODBC from the client without connecting to SPSS Modeler Server, then install the IBM SPSS Data Access Pack. See the installation instructions available online [here](#) for Windows or [here](#) for UNIX.
-

Optional components

Additional IBM® SPSS® Collaboration and Deployment Services components are available for installation. To install any of the following, click the links to see the installation instructions.

- [IBM SPSS Collaboration and Deployment Services - Essentials for Python](#)
- [IBM SPSS Collaboration and Deployment Services Remote Scoring Server](#)
- [IBM SPSS Collaboration and Deployment Services Remote Process Server](#)

Note: The Remote Scoring Server and the Remote Process Server are usually installed on different machines, and they are typically not installed on the same machine as the IBM SPSS Collaboration and Deployment Services Repository (the repository already contains the functionality). For more information about these components, see the IBM SPSS Collaboration and Deployment Services documentation.

Post-installation steps

After installation, perform the following required configuration steps. Depending on your environment, additional configuration may be required.

- [IBM SPSS Collaboration and Deployment Services Server components \(Server 1\)](#)
- [IBM SPSS Modeler Server components \(Server 2\)](#)
- [Client components](#)

- [Other](#)
-

IBM SPSS Collaboration and Deployment Services Server components (Server 1)

1. Start the repository server. For instructions, see [here](#).
 2. Verify the repository is running by accessing browser-based IBM® SPSS® Deployment Manager. Navigate to the login page at <http://<repository host>/<port number>/security/login> and specify the admin login credentials that were specified during repository configuration.
 3. For additional information about post-installation steps for IBM SPSS Collaboration and Deployment Services, see the documentation [here](#).
-

IBM® SPSS Modeler Server components (Server 2)

1. To configure the SPSS® Modeler Server, see [this documentation](#) (for Windows) or [this documentation](#) (for UNIX) and follow the instructions in these sections:
 - *Checking the Server Status*
 - *Connecting End Users*
 - *IBM SPSS Data Access Pack Technology*
 2. Use the IBM SPSS Data Access Pack to configure an ODBC driver that points to your database. See [this PDF](#) (for Windows) or [this PDF](#) (for UNIX).
-

Client components

Perform the following configuration on the client machine(s). The first 4 steps use the IBM® SPSS® Deployment Manager client software.

1. Open Deployment Manager client, go to the Content Explorer tab, and create a new content server connection if you have not done so already. See the instructions [here](#). After creating the connection, double-click it and log on (you can use the *admin* account password that was specified during Repository installation).
 2. Create a new IBM SPSS Modeler Server definition and associated credentials definition.
For additional instructions about creating server definitions and credentials, see the online documentation [here](#).
 3. Go to the Server Administration tab and create an administered server connection if you have not done so already. Then create users and groups as required for your deployment and assign them roles. For instructions on creating groups and assigning roles, see the documentation [here](#).
 4. For each SPSS Modeler Client, create a connection to the IBM SPSS Modeler Server hosted on *Server 2*, using [these](#) instructions.
 5. Open SPSS Modeler Client and create a connection to the IBM SPSS Collaboration and Deployment Services Repository hosted on *Server 1*. See the instructions available [here](#).
 6. If you installed IBM SPSS Collaboration and Deployment Services - Essentials for Python, see the instructions [here](#) to verify the installation.
-

Other

Administration consoles

- To facilitate easier server administration and configuration, IBM® SPSS® Deployment Manager client includes the IBM SPSS Modeler Administration Console. This provides for a single place to administer all SPSS Modeler servers.
The console can be used to perform tasks such as starting and stopping servers and changing configuration settings.
-

Passport Advantage part numbers

The following table lists the [Passport Advantage](#) part number for each eAssembly that makes up the IBM® SPSS® Modeler Gold offering. For a complete list of all eImages included in each of the eAssemblies listed below, see the SPSS Modeler Gold sections of the table at the end of the IBM SPSS Modeler 18.4 [download document](#).

Table 1. Passport Advantage parts

SPSS component	Part number
IBM SPSS Modeler Gold 18.4 Multiplatform Multilingual eAssembly ¹	G06C3ML ¹
IBM SPSS Modeler Gold Keyless 18.4 Multiplatform eAssembly ²	G06C4ML ²
IBM SPSS Modeler Server Gold 18.4 Microsoft Windows Multilingual eAssembly	G06C8ML
IBM SPSS Modeler Server Gold 18.4 Linux x86-64 Multilingual eAssembly	G06C7ML
IBM SPSS Modeler Server Gold 18.4 Linux on z Systems Multilingual eAssembly	G06C9ML
IBM SPSS Modeler Server Gold 18.4 Linux on System p LE Multilingual eAssembly	G06CBML
IBM WebSphere Application Server V9.0.5 for IBM SPSS Modeler Server Gold 18.4 Multiplatform Multilingual eAssembly	G06CCML
IBM Db2 Standard Edition V11.5 for IBM SPSS Modeler Gold 18.4 Multiplatform Multilingual eAssembly	G06CDML
IBM SPSS Data Access Pack V8.1.1 Multiplatform English	M06RGEN
IBM SPSS Modeler Gold Client Keyless 64-bit 18.4 Microsoft Windows Multilingual	M06RCML
IBM SPSS Modeler Gold Client Keyless 18.4 Mac OS Multilingual	M06RDML

¹ This eAssembly will be used if you intend to install *concurrent* user clients (and, therefore, need to supply license keys). If you only have entitlements to concurrent users, you will not see the *IBM SPSS Modeler Gold Keyless 18.4 Multiplatform eAssembly* download.

² This eAssembly will be used if you intend to install *authorized* user clients (which do not need to be licensed via keys). You will only see this download option if you have authorized users (or both authorized users and concurrent users).

IBM SPSS Modeler considerations for GDPR readiness

Information about features of IBM® SPSS® Modeler that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness.

For PID(s): 5725-A64, 5725-A65

Notice:

This document is intended to help you in your preparations for GDPR readiness. It provides information about features of SPSS Modeler that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness. This information is not an exhaustive list, due to the many ways that clients can choose and configure features, and the large variety of ways that the product can be used in itself and with third-party applications and systems.

Clients are responsible for ensuring their own compliance with various laws and regulations, including the European Union General Data Protection Regulation. Clients are solely responsible for obtaining advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulations that may affect the clients' business and any actions the clients may need to take to comply with such laws and regulations.

The products, services, and other capabilities described herein are not suitable for all client situations and may have restricted availability. IBM does not provide legal, accounting, or auditing advice or represent or warrant that its services or products will ensure that clients are in compliance with any law or regulation.

Table of contents

1. [GDPR](#)
2. [Product Configuration for GDPR](#)
3. [Data Life Cycle](#)
4. [Data Collection](#)
5. [Data Storage](#)
6. [Data Access](#)
7. [Data Processing](#)
8. [Data Deletion](#)
9. [Data Monitoring](#)
10. [Responding to Data Subject Rights](#)

GDPR

General Data Protection Regulation (GDPR) has been adopted by the European Union and will apply from May 25, 2018.

Why is GDPR important?

GDPR establishes a stronger data protection regulatory framework for processing of personal data of individuals. GDPR brings:

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Potential for significant financial penalties for non-compliance
- Compulsory data breach notification

Read more about GDPR:

- [EU GDPR Information Portal](#)
- [ibm.com/GDPR web site](#)

Product configuration - considerations for GDPR readiness

Offering Configuration

The following sections provide considerations for configuring SPSS Modeler to help your organization with GDPR readiness.

To configure SPSS Modeler such that it supports GDPR readiness, IBM recommends that you:

- Activate the encryption of data in motion (data that is transferred over a data network between SPSS Modeler client applications and SPSS Modeler Server; data between SPSS Modeler Server and external execution servers such as IBM SPSS Collaboration and Deployment Services Server or IBM SPSS Analytic Server).
- Do not share passwords for the operation of SPSS Modeler between multiple persons.
- Apply a firewall in front of the offering to avoid security vulnerabilities.

Data Life Cycle

GDPR requires that personal data is:

- Processed lawfully, fairly and in a transparent manner in relation to individuals.
- Collected for specified, explicit and legitimate purposes.
- Adequate, relevant and limited to what is necessary.
- Accurate and, where necessary, kept up to date. Every reasonable step must be taken to ensure that inaccurate personal data are erased or rectified without delay.
- Kept in a form which permits identification of the data subject for no longer than necessary.

What is the end-to-end process through which personal data go through when using SPSS Modeler?

What types of data flow through SPSS Modeler?

SPSS Modeler is a powerful predictive analytics platform that is designed to bring predictive intelligence to decisions made by individuals, groups, systems, and your enterprise. SPSS Modeler scales from desktop deployments to integration with operational systems to provide you with a range of advanced algorithms and techniques. The scalable client-server architecture enables users to access everything from flat files to big data environments. SPSS Modeler provides powerful data mining abilities and enables users to visualize each step of the data mining process.

SPSS Modeler doesn't decide what data is sensitive and what is not sensitive. The data asset to be stored by SPSS Modeler can include personal data or not, which is fully dependent on your purpose. Also, you have total control of how your data is handled, used, and stored by SPSS Modeler. Only for debugging and auditing purposes, the personal information that Modeler collects includes the user ID used to log in to SPSS Modeler, the IP address for the connection, and the database connection information.

Where in the process?

- Authentication: All operations to SPSS Modeler must be authenticated. When users connect to SPSS Modeler Server via SPSS Modeler client, users must log on to the server with an authenticated user ID on the operating system where the server is running. SPSS Modeler will prompt users to input the password once the connection is closed (for example, when SPSS Modeler client is restarted). SPSS Modeler doesn't store passwords.
- Logging: The files `server_logging.log`, `server_sessions.log`, and `server_tracing_<sequence>.log` on the server application side are used to collect usage data as logs for diagnostic purposes. SPSS Modeler administrators can enable/disable the logs or change the log level based on his/her needs. This is fully controlled by the authorized administrator. The file `client_logging.log` on the client application side is used to collect usage data as logs for diagnostic purposes. SPSS Modeler client application users can enable/disable the logs or change the log level based on his/her needs. This is fully controlled by the authorized user.
- Configuration: SPSS Modeler configuration is stored in a configuration file that is restricted by file-level authentication and encryption.

For what purpose?

SPSS Modeler software doesn't store business data by itself. SPSS Modeler can import data from a data source or export data to a data destination. The data source and data destination might be: file, database, IBM SPSS Collaboration and Deployment Services, or IBM SPSS Analytic Server.

The business data are mainly used by SPSS Modeler for data mining purposes. The operational data that is captured in the logs and files will be used mainly for SPSS Modeler troubleshooting purposes.

Personal data used for online contact with IBM

SPSS clients can submit online comments/feedback/requests to contact IBM about SPSS subjects in a variety of ways, primarily:

- Public comments area on pages in the SPSS community on IBM developerWorks®
- Public comments area on pages of SPSS product documentation in IBM Documentation
- Public comments in the SPSS space of dWAnswers
- Feedback forms in the SPSS community

Typically, only the client name and e-mail address are used, to enable personal replies for the subject of the contact, and the use of personal data conforms to the [IBM Online Privacy Statement](#).

Data Collection

When assessing your use of SPSS Modeler and the requirements of GDPR, you should consider the types of personal data which in your circumstances are passing through SPSS Modeler.

Account data encompasses several scenarios. The user name is collected in the log files server_sessions.log and server_logging.log on the remote SPSS Modeler Server. IBM SPSS Analytic Server logon information is collected in the file client_logging.log if you connect to IBM SPSS Analytic Server directly from SPSS Modeler client. The database and other systems' logon information will only be collected in the files server_logging.log or server_tracing_**.log. Client data isn't collected in the logs.

Data Storage

The following Data Storage mechanisms are used by SPSS Modeler, which users may wish to consider when assessing their GDPR readiness.

- Storage of account data: Account data can be securely stored in a SPSS Modeler working file (.str) which is restricted by file authentication and encryption, or repository authentication and encryption (IBM SPSS Collaboration and Deployment Services). SPSS Modeler doesn't store user passwords in any way.
- Storage of client data: For client data that's stored in database, it requires database level authentication and encryption. For client data stored on the file system, it requires file system access control. For client data stored in other collaboration systems (for example, IBM SPSS Collaboration and Deployment Services or IBM SPSS Analytic Server), it requires access control of the corresponding system.

Data Access

- In SPSS Modeler, access control to personal data can be achieved by operating system permissions to files generated by SPSS Modeler (for example, stream files, log files, and preferences files).
- Roles and access rights: SPSS Modeler doesn't manage any roles or access rights by itself. Access control can be achieved by the operating system or other collaboration systems (database, IBM SPSS Collaboration and Deployment Services, or IBM SPSS Analytic Server).
- Separation of duties: SPSS Modeler does not manage any duties separation by itself. Access control can be achieved by the operating system or other collaboration systems (database, IBM SPSS Collaboration and Deployment Services, or IBM SPSS Analytic Server).
- Administrators: SPSS Modeler doesn't have any administrators, and access control can be achieved by the operating system or other collaboration systems (database, IBM SPSS Collaboration and Deployment Services, or IBM SPSS Analytic Server).
- Activity logs: The logs include info about who connects to the SPSS Modeler Server, when to start the process, and when to close the session. This information is stored in SPSS Modeler log files, whose access is restricted by file level authentication and encryption.

Data Processing

- Encryption in motion: SSL or TLS is supported by SPSS Modeler when SPSS Modeler is connecting to other systems (database, IBM SPSS Collaboration and Deployment Services, or IBM SPSS Analytic Server).
- Encryption at rest: When a SPSS Modeler stream (.str) is saved, you can select an option to encrypt the file.

Data Deletion

Right to Erasure

Article 17 of the GDPR states that data subjects have the right to have their personal data removed from the systems of controllers and processors - without undue delay - under a set of circumstances.

Data Deletion Characteristics

- Client data deletion: Client data can be deleted via features in the SPSS Modeler user interface.
- Account data deletion: Account data can be deleted via features in the SPSS Modeler user interface.

Data Monitoring

You should regularly test, assess, and evaluate the effectiveness of your technical and organizational measures to comply with GDPR. These measures should include ongoing privacy assessments, threat modeling, centralized security logging, and monitoring, among others.

The SPSS Modeler logs allow clients to track who connects to the SPSS Modeler Server, when, start of session, start of process, and close of session.

Responding to Data Subject Rights

- Right to Access: SPSS Modeler doesn't manage any roles or access rights by itself. This is performed via the operating system or other collaboration systems (database, IBM SPSS Collaboration and Deployment Services, or IBM SPSS Analytic Server).
- Right to Modify: SPSS Modeler has the capability to work with client data. Only the authorized user can modify or correct the data through SPSS Modeler client.
- Right to Restrict Processing: SPSS Modeler client can stop a running stream at any time to force the server to stop processing the client data.
- Right to Object: When objects (all kinds of outputs and models) are generated during the running of SPSS Modeler streams, the client user can delete these objects directly. When objects (such as database, file system, and others) are exported from SPSS Modeler, only the authorized user can access and delete them.
- Right to Be Forgotten: Same as Right to Object above.
- Right to Data Portability: SPSS Modeler supports data portability to a great extent. You can export models to PMML and used them outside of SPSS Modeler. SPSS Modeler streams can be saved and deployed to other systems. Model output can be exported to several file types such as flat file, IBM SPSS Statistics format file, SAS format file, database tables etc.